Adaptive and Blind Regression for Mobile Crowd Sensing

Shan Chang, *Member, IEEE*, Chao Li, Hongzi Zhu, *Member, IEEE*, and Hang Chen

Abstract—In mobile crowd sensing (MCS) applications, a public model of a system is expected to be derived from observations collected by mobile device users, through regression modeling. For example, a model describing the relationship between running speed, heart rate, height, and weight of runner can be constructed using MCS data collected from wristbands. Unique features of MCS data bring regression 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 nonstationary and opportunistically, calling for an adaptive model updating mechanism. Last, mobile devices are resource-constrained, posing an urgent demand for lightweight regression. We propose an adaptive and blind regression scheme. The core idea is first to select an optimal ‘safe’ subset of observations locally stored over all participants, such that the inconsistency between the subset and the corresponding regression model is minimized, and as many observations as possible are included. Then, based on the resulted regression model, more observations are checked and selected to refine the model. With observations constantly coming, newly selected ‘safe’ observations are used to make the model updated adaptively. To preserve data privacy, *one-time pad masking* and *blocking scheme* are integrated.

Index Terms—Mobile crowd sensing, blind regression, adaptive model updating, outlier, opportunistic sensing, non-stationary

1 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 condition forecasting [3], [4], etc.

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. For example, a fuel-saving navigation service relies on MCS data collected from smart vehicles to construct the linear model between fuel cost and routing decision. Or, a blood pressure predicting service relies on MCS data collected from smart sphygmomanometers to construct a model between hour of sleep, age, salt intake and blood pressure. 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 task 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) Adaptive 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 adaptive 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 the 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

- *S. Chang, C. Li, and H. Chen are with the School of Computer Science & Technology, Donghua University, Shanghai 201620, P. R. China. E-mail: changshan@dhu.edu.cn, {chaoli, chenhang}@mail.dhu.edu.cn, hongzi@cs.sjtu.edu.cn.*
- *H. Zhu is with the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200000, P. R. China.*

Manuscript received 22 Jan. 2019; revised 9 June 2019; accepted 10 July 2019. Date of publication 29 July 2019; date of current version 1 Oct. 2020. (Corresponding author: Hongzi Zhu.) Digital Object Identifier no. 10.1109/TMC.2019.2931341

1536-1233 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

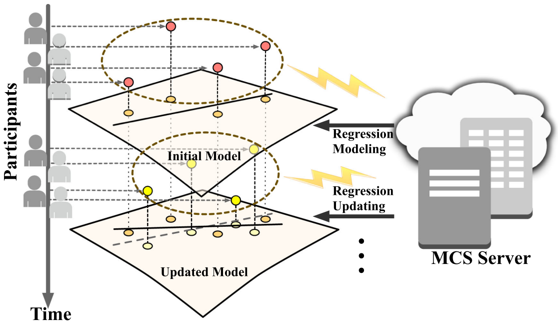


Fig. 1. Illustration of Lotus used in mobile crowd sensing applications.

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 sensory data are collected with mobile devices of unprofessional participants who can communicate with the MCS server via WiFi and 3G/4G. A regression estimator is proposed, called *Lotus*, that can be implemented as a set of protocols running between the server and mobile devices of participants. To deal with opportunistic and non-stationary observations, as illustrated in Fig. 1, Lotus estimates a model which can update adaptively with newly collected observations periodically. More specifically, a two-step estimation is made for building an original regression model. First, we propose an optimal-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 determine. Considering finding such an optimal-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, 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. We estimate a rough global regression model based on the final 'safe' subset. Second, we use this rough model to examine the validity (outlyingness) of local observations outside the optimal 'safe' subset. Those observations which are considered valid will be utilized to refine the global estimate. Moreover, the model updates once enough new observations are available. During model updating, a new 'safe' subset of fresh observations is first determined in the same way as adopted in estimating the original model to combine with the current model. With the combined model, which is considered as the new rough model, each participant re-examines the validity of new observations and old observations that have already been involved in the original model as well, for the purpose of updating the original model.

We emphasize that Lotus will not require participants to share their raw observations with any others including the server because of the constraint of privacy-preservation, which poses a challenge to achieving reliable model estimates on contaminated observations (without acquiring the raw data, it is hard to identify and further remove outliers, making 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, aggregation of local observations is designed to leverage the additive property of multiplication of partitioned matrices. As a result, new valid observations can be added in the new refined model while out-of-date observations can be removed from the new model in an incremental way.

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 improved over time. Last but not least, Lotus is a lightweight protocol tailored for mobile devices. We conduct intensive analysis to prove that Lotus can protect private data from being spied and can defend against collusion attacks. We evaluate the performance of Lotus through both trace-driven simulations and real-world implementation. The results demonstrate the effectiveness and robustness of Lotus to model updating and in the presence of outliers even under a ratio of 40 percent.

The remainder of this paper is organized as follows. In Section 2, we introduce problem formulation and preliminaries. Section 3 describes the design of Lotus. In Section 4, we presents the analysis of computational complexity and the security. Section 5 shows the performance evaluation. In Section 6, we elaborate the prototype implementation of Lotus. We review related work in Section 7. Section 8 concludes the paper.

2 Problem Formulation and Preliminaries

2.1 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.

2.2 Multivariate Linear Regression

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

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

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. For example, the rent of a house (dependent variable y) depends on six (p) factors, i.e., the neighborhood (x_1), size (x_2), number of rooms (x_3), distance of nearest station (x_4), distance of nearest shopping mall (x_5), and attached facilities (x_6). The jth 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

$$\begin{aligned} &\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \\ &\boldsymbol{u} = \sum_{i=1}^{m} (\boldsymbol{x}^{(i)})^T \boldsymbol{x}^{(i)}, \quad \boldsymbol{v} = \sum_{i=1}^{m} (\boldsymbol{x}^{(i)})^T \boldsymbol{y}^{(i)} \\ &\widehat{\boldsymbol{\beta}} = (\boldsymbol{u})^{-1}\boldsymbol{v}. \end{aligned} \tag{2}$$

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 careful, since abuse of aggregated results is vulnerable to observations recovery attacks.

2.3 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 percent; 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 percent 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.*

2.4 Adaptive Regression with Non-Stationary Observations

In MCS, a regression model is updated 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* adaptive 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.

2.5 Mahalanobis Distance

The Mahalanobis distance [13] 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:

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

$\mathbf{o}_i$ 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.

2.6 Design Goals and Challenges

We aim to develop a practical method to address the problems of Definitions 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 updating 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 adaptive 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

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

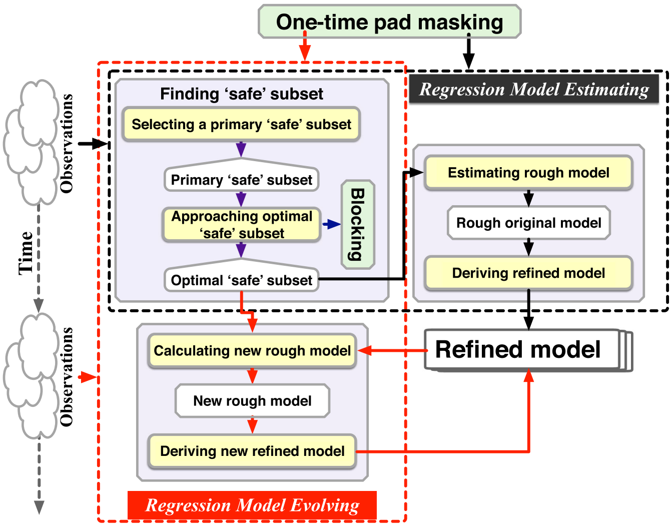


Fig. 2. Overview of lotus.

(e.g., homomorphic encryption), which induce high transmission and computation cost.

3 Design of Lotus

3.1 Overview

Lotus incorporates two techniques, i.e., blind regression model estimating and blind regression updating, as illustrated in Fig. 2. Notice that, during regression, we develop a pair of lightweight secure calculation (aggregation and comparison) schemes, which is intended to ensure the 'blind' property.

Blind Regression Model Estimating. This technique is used to derive an original regression model according to a set of observations, which has two steps. *1) Estimating rough model with optimal 'safe' subset:* considering that outlier ratio is smaller than 50 percent, we wish to choose a 'safe' subset with half observations, which satisfies that the rough model estimated accordingly achieves minimum RSS, implying that observations in selected 'safe' subset obey quite the same trend. In other words, a 'tight' and 'safe' regression model can be made and used as a rough original model. However, finding such a subset is proven NP-hard. Thus we develop a two-stage hill-climbing approach to achieve an approximation of optimal solution. First, we *select a primary 'safe' subset,* to initializing rough model as the starting point of hill-climbing. 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 check. The first half observations are considered as a primary 'safe' subset, and the corresponding regression model is intended as the starting point. Then, we introduce a *swap algorithm* to *approach the optimal 'safe' subset* for estimating the rough original model, where observations inside and outside of the 'safe' subset will be swapped iteratively, such that the RSS of the corresponding model can be minimized. *2) Deriving refined model:* by checking the data quality again using the rough model estimated, more valid local observations can be found and used to refine the rough model, ultimately resulting in the original model.

Blind Regression Updating. This technique performs two functions to achieve an updated model estimate. *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 have 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.

Secure Aggregation and Comparison. In the above stages, in order to protect data privacy, each participant masks their local aggregated results used for model estimating with particular random values, i.e., *one-time pad masking*. 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, which enables that residuals of observations can be compared in a secure manner, in the procedure of optimizing 'safe' subset.

3.2 Blind Regression Model Estimating

3.2.1 Estimating Rough Global Model with Optimized 'Safe' Subset

In our design, we wish to use a subset with minimum risk of including dirty observations as the 'safe' subset. Thus, we give the following definition:

Definition 4. *The problem of* finding optimal 'safe' subset *is referred to as, given a set $\mathcal{O}$ of n observations with p (constant) independent and 1 dependent variables, i.e., an observation $\boldsymbol{o_i} = (\boldsymbol{x_i}, y_i), (i = 1, \ldots, n)$, where $\boldsymbol{x_i}$ is a p dimension vector, select $\lceil \frac{n}{2} \rceil$ observations to fit a regression model, to minimize corresponding RSS.*

The reason for forming a size $\lceil \frac{n}{2} \rceil$ 'safe' subset is two-folds. On one side, if the subset has less than half observations, it is possible that all observations are dirty. On the other side, if more than half observations are selected, the risk of including outliers is increased. However, the problem of finding optimal 'safe' subset is very hard. We have the following Theorems.

Theorem 1. *The problem of finding optimal 'safe' subset defined in Definition 4 is NP-hard.*

Proof. Suppose that we apply LS estimator to all observations in set $\mathcal{O}$, then the corresponding RSS, i.e., $R_{ss} = \sum_{i=1}^{n} (e_i)^T \boldsymbol{e}_i$, can be minimized, where $e_i = y_i - \boldsymbol{x}_i \widehat{\boldsymbol{\beta}}$, and $\widehat{\boldsymbol{\beta}}$ is the regression model estimated. We can reformulate R_{ss} as

$$R_{ss} = Y^T Y - \widehat{\boldsymbol{\beta}}^T X^T Y,$$

where $X = [\boldsymbol{x}_1, \ldots, \boldsymbol{x_n}]^T$ and $Y = [y_1, \ldots, y_n]^T$.

Remember that $\widehat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T Y$, then if we select $\lceil \frac{n}{2} \rceil$ observations from set $\mathcal{O}$ to fit an LS regression model $\bar{\boldsymbol{\beta}}$, it can be obtained by using

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

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

where W is an $n \times n$ diagonal matrix , such that

$$w_{i,i} = \begin{cases} 1 & \text{if observation } \boldsymbol{o_i} \text{ is selected} \\ 0 & \text{otherwise.} \end{cases}$$

Meanwhile, we have the corresponding RSS, i.e.,

$$\begin{aligned} \bar{R}_{ss} &= Y^TWY - \bar{\boldsymbol{\beta}}^T X^TWY \\ &= Y^TWY - [(X^TWX)^{-1}X^TWY]^T X^TWY \\ &= Y^TWY - Y^TW^TX[(X^TWX)^{-1}]^T X^TWY. \end{aligned}$$

Then the problem in Definition 4 can be expressed as the following non-linear 0-1 programming problem

$$\textbf{Minimize } Y^TWY - Y^TW^TX[(X^TWX)^{-1}]^TX^TWY$$

$$\textbf{Subject to } \sum_{i=1}^{n} w_{i,i} = \left\lceil \frac{n}{2} \right\rceil, \quad w_{i,i} \in \{0, 1\}.$$

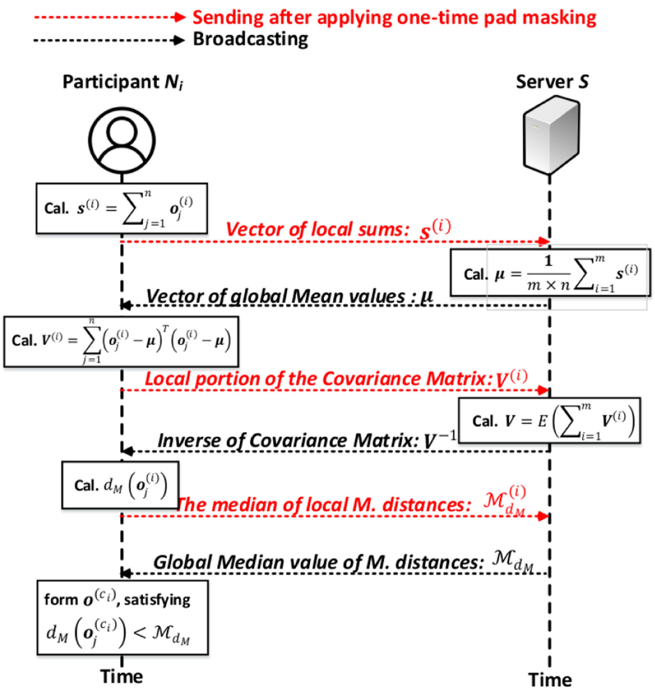
It is well known that non-linear 0-1 programming problem is NP-hard [14], [15], This completes the proof. □

Therefore, we use a greedy *hill-climbing algorithm* to approximate the optimum solution, where an iterative algorithm that starts with a Mahalanobis distance-based primary 'safe' subset, then attempts to find a better 'safe' subset by making an incremental change (by conducting observation swapping) to the subset.

Selecting Primary 'Safe' Subset: In this step, we select a subset of observations, used as the starting point of hill climbing, in procedure of Approaching Optimal 'Safe' Subset. Hill climbing starts from an arbitrary solution to a problem, then tries to find a better solution by making an incremental change to the current one, and so on until no further improvements can be made. It is critical to find a good starting point (solution) close to the global maxima (best solution) as much as possible, in order to prevent the search stop at local maxima, and to reduce the number of searches as well. We aim to find a subset which is presumably free of outliers as the starting point, since a subset contaminated with outliers may lead to a final solution composed entirely of outliers. In order to detect outliers, it is necessary to take into consideration the shape of the data set. For example, given an elliptical shape cloud of 2-dimension data points, some points are closer to the center than others, however, we cannot conclude that those large distance points (in terms of the classical Euclidean distance) belong less to the cloud than short distance ones, as this is part of the underlying pattern of the data point distribution. Hence, instead of the Euclidean distance, we utilize Mahalanobis distance, which takes into account the shape of the observations.

Suppose $N_i \in \mathbb{N}$ 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 jth 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 represented 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 a primary 'safe' subset. Figure 3 depicts the protocol between the server and each participant.

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 primary 'safe' subset.



Sending after applying one-time pad masking
Broadcasting
Participant N_i
Server S
Cal. $s^{(i)} = \sum_{j=1}^{n} o_j^{(i)}$
Vector of local sums: $s^{(i)}$
Cal. $\mu = \frac{1}{m \times n} \sum_{i=1}^{m} s^{(i)}$
Vector of global Mean values : μ
Cal. $V^{(i)} = \sum_{j=1}^{n} (o_j^{(i)} - \mu)^T (o_j^{(i)} - \mu)$
Local portion of the Covariance Matrix: $V^{(i)}$
Cal. $V = E\left(\sum_{i=1}^{m} V^{(i)}\right)$
Inverse of Covariance Matrix: V^{-1}
Cal. $d_M\left(o_j^{(i)}\right)$
The median of local M. distances: $\mathcal{M}_{d_M}^{(i)}$
Global Median value of M. distances: $\mathcal{M}_{d_M}$
form $o^{(c_i)}$, satisfying $d_M\left(o_j^{(c_i)}\right) < \mathcal{M}_{d_M}$
Time
Time

Fig. 3. Protocol of finding primary '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 broadcasts $\boldsymbol{\mu}$ to $\mathbb{N}$.
- *Calculating* $\boldsymbol{V}^{-1}$*:* 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

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

  then calculates and broadcasts $\boldsymbol{V}^{-1}$, the inverse of $\boldsymbol{V}$, to $\mathbb{N}$.
- *Calculating* d_M*:* each N_i computes $d_M(\boldsymbol{o}_j^{(i)})$ according to (3).
- *Estimating the global median of* d_M*:* each N_i sends the local median of $d_M(\boldsymbol{o}^{(i)})$, i.e., $\mathcal{M}_{d_M}^{(i)}$, to S. S sorts all $d_M(\boldsymbol{o}_j^{(i)})$ received, and broadcasts the median $\mathcal{M}_{d_M}$ (which is the global median) to $\mathbb{N}$.
- *Preparing local primary '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 $\boldsymbol{o}^{(c_i)} = \{\boldsymbol{o}_1^{(c_i)}, \boldsymbol{o}_2^{(c_i)}, \ldots, \boldsymbol{o}_{\xi_i}^{(c_i)}\}$. Notice that the union of local primary 'safe' subset, i.e., $\boldsymbol{o}^{(\mathbf{c})} = \bigcup_{i=1}^{m} \boldsymbol{o}^{(c_i)}$, is the global primary subset we needed.

According to the primary 'safe' subset obtained, S cooperates with participants on initializing rough model without knowing $\boldsymbol{o}^{(c)}$.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

More specifically, N_i computes $(\boldsymbol{x}^{(c_i)})^T(\boldsymbol{x}^{(c_i)})$, $(\boldsymbol{y}^{(c_i)})^T(\boldsymbol{y}^{(c_i)})$ and $(\boldsymbol{x}^{(c_i)})^T\boldsymbol{y}^{(c_i)}$ using its local primary ‘safe’ subset, and submits the results to S. 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) (i.e., $\widehat{\boldsymbol{\beta}}_0 = (\boldsymbol{u}_0^{(c)})^{-1}\boldsymbol{v}_0^{(c)}$). Furthermore, S calculates the corresponding RSS, i.e., $R_{ss_{(0)}}$, which equals to $\boldsymbol{w}_0^{(c)} - (\widehat{\boldsymbol{\beta}}_0)^T\boldsymbol{v}_0^{(c)}$, and will be used in the next stage.

In the above procedure, N_i needs to provide a number of aggregated results, e.g., $(\boldsymbol{x}^{(c_i)})^T(\boldsymbol{x}^{(c_i)})$ and $(\boldsymbol{x}^{(c_i)})^T(\boldsymbol{y}^{(c_i)})$, which closely relate to its raw observations $\boldsymbol{o}^{(c_i)}$. However, disclosing those aggregates may cause privacy problem if the number of variables in $\boldsymbol{o}_c^{(i)}$ is no more than that of the equations built from them. Thus, sharing those aggregates directly is prohibited.

Considering, according to the protocol, the server adds up each kind of aggregated values from all participants together for further processing, and doesn’t care about individual ones, we introduce a *one-time pad masking* technique for calculating the summation of aggregated values without disclosing individual ones, which will be explained in Section 3.4.1.

Remarks. The number of observations used for building a regression model should be no less than $2p+3$, which prevents the primary ’safe’ subset $\boldsymbol{o}^{(\boldsymbol{c})}$ from being recovered by the server S (See *Theorem 2* and the proof in Section 4.2.1).

Approaching Optimal ’Safe’ Subset: we use the primary ’safe’ subset $\boldsymbol{o}^{(\boldsymbol{c})}$, leading to $\widehat{\boldsymbol{\beta}}_0$, as the starting point of hill climbing, and incrementally improve ’safe’ subset towards ’optimal’ (as defined in Definition 4). To this end, we check the residuals of remaining observations, and give the following definition:

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

Optimizing ’safe’ subset refers to adjusting members of subset such that a corresponding model with minimum RSS can be achieved. The MRE of an observation group implies the average residual induced by each observation in it. For the purpose of reducing the RSS, those observations outside the primary ’safe’ subset (named ’uncertain’) 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. As a result, the primary ’safe’ subset is updated towards shrinking MRE as well as RSS, and a new regression model can be estimated according to it. In detail, the following *observation-swapping protocol* is performed between each N_i and the server S, (illustrated in Fig. 4),

- *Announcing current model and corresponding MRE:* S calculates MRE of current model $\widehat{\boldsymbol{\beta}}_0$, denoted as $Mre_{(0)}$, by using

$$Mre_{(0)} = \sqrt{\frac{\boldsymbol{w}_0^{(c)} - (\widehat{\boldsymbol{\beta}}_0)^T\boldsymbol{v}_0^{(c)}}{n}},$$

and broadcasts $\widehat{\boldsymbol{\beta}}_0$ and $Mre_{(0)}$ to all participants.

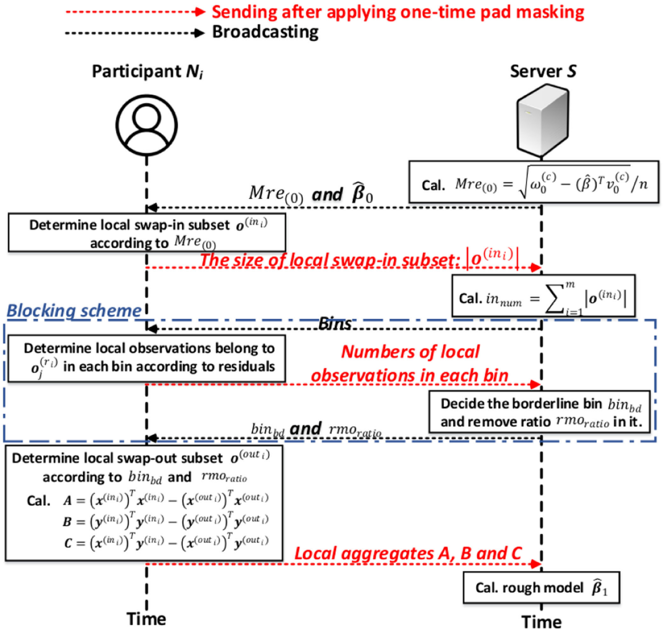

Sending after applying one-time pad masking
Broadcasting
Participant N_i
Server S
Cal. $Mre_{(0)} = \sqrt{\omega_0^{(c)} - (\hat{\beta})^T v_0^{(c)}}/n$
$Mre_{(0)}$ and $\widehat{\beta}_0$
Determine local swap-in subset $\boldsymbol{o}^{(in_i)}$ according to $Mre_{(0)}$
The size of local swap-in subset: $|\boldsymbol{o}^{(in_i)}|$
Cal. $in_{num} = \sum_{i=1}^m |\boldsymbol{o}^{(in_i)}|$
Blocking scheme
Bins
Determine local observations belong to $\boldsymbol{o}_j^{(r_i)}$ in each bin according to residuals
Numbers of local observations in each bin
Decide the borderline bin bin_{bd} and remove ratio rmo_{ratio} in it.
bin_{bd} and rmo_{ratio}
Determine local swap-out subset $\boldsymbol{o}^{(out_i)}$ according to bin_{bd} and rmo_{ratio}
Cal. $A = (\boldsymbol{x}^{(in_i)})^T\boldsymbol{x}^{(in_i)} - (\boldsymbol{x}^{(out_i)})^T\boldsymbol{x}^{(out_i)}$
$B = (\boldsymbol{y}^{(in_i)})^T\boldsymbol{y}^{(in_i)} - (\boldsymbol{y}^{(out_i)})^T\boldsymbol{y}^{(out_i)}$
$C = (\boldsymbol{x}^{(in_i)})^T\boldsymbol{y}^{(in_i)} - (\boldsymbol{x}^{(out_i)})^T\boldsymbol{y}^{(out_i)}$
Local aggregates A, B and C
Cal. rough model $\widehat{\beta}_1$
Time
Time

Fig. 4. Protocol of approaching optimal ‘safe’ subset.

- *Deciding local swap-in subset:* each 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 $Mre_{(0)}$ are formed a swap-in subset $\boldsymbol{o}^{(in_i)}$, and N_i submits the size of $\boldsymbol{o}^{(in_i)}$, i.e.,$|\boldsymbol{o}^{(in_i)}|$ to S by using one-time pad masking.
- *Determining global swap-out threshold of residual:* 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.
- *Updating local ’safe’ subset:* 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 $e_j^{(c_i)} > e_{trsd}$.
- *Revising model using updated ’safe’ subset:* each 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 to S, by using one-time pad masking. S computes $\boldsymbol{u}_1^{(c)}$ as follow,

$$\begin{aligned}\boldsymbol{u}_1^{(c)} &= \boldsymbol{u}_0^{(c)} + \sum_{i=1}^m ((\boldsymbol{x}^{(in_i)})^T\boldsymbol{x}^{(in_i)} - (\boldsymbol{x}^{(out_i)})^T\boldsymbol{x}^{(out_i)}) \\ &= \boldsymbol{u}_0^{(c)} + \sum_{i=1}^m (\boldsymbol{x}^{(in_i)})^T\boldsymbol{x}^{(in_i)} - \sum_{i=1}^m (\boldsymbol{x}^{(out_i)})^T\boldsymbol{x}^{(out_i)}.\end{aligned}$$

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 $Rss_{(1)} = \boldsymbol{w}_1^{(c)} - (\widehat{\boldsymbol{\beta}}_1)^T\boldsymbol{v}_1^{(c)}$.

The above procedure will be carried out iteratively until no ‘uncertain’ and ‘safe’ observations can be swapped in and out of the ultimate (optimal) ’safe’ subset. The

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

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., by using Garble Circuit, which however induces high computation and communication costs. We design a light-weight and effective *blocking* scheme by loosening the accuracy of $\boldsymbol{o}^{(out)}$, i.e., $\bigcup_{i=1}^{m}\boldsymbol{o}^{(out_i)}$, slightly, and explain the scheme in Section 3.4.2.

3.2.2 Deriving Refined Global Model

After achieving the rough model $\widehat{\boldsymbol{\beta}}_{safe}$ (i.e., $(\boldsymbol{u}^{(c)})^{-1}\boldsymbol{v}^{(c)}$), 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 its remaining subset $\boldsymbol{o}^{(r_i)}$ (i.e., 'uncertain') is tested in accordance with $\widehat{\boldsymbol{\beta}}_{safe}$. Those observations fitting $\widehat{\boldsymbol{\beta}}_{safe}$ well will be considered as 'safe', and be utilized for refining $\widehat{\boldsymbol{\beta}}_{safe}$, which leads to an original estimate $\widehat{\boldsymbol{\beta}}_{org}$.

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

$$z_j^{(r_i)} = |e_j^{(r_i)}|/RMse. \tag{4}$$

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 rough regression model. We emphasize that $(\boldsymbol{x}^{(f_i)})^T\boldsymbol{x}^{(f_i)}$ and $(\boldsymbol{x}^{(f_i)})^T\boldsymbol{y}^{(f_i)}$ are computed locally, and are submitted to S by using masking.

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

3.3 Blind Regression Updating

Under the current estimate $\widehat{\boldsymbol{\beta}}_{org}$, 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., $\widetilde{\boldsymbol{o}}^{(l)} = \{\widetilde{\boldsymbol{o}}_1^{(l)}, \widetilde{\boldsymbol{o}}_2^{(l)}, \ldots, \widetilde{\boldsymbol{o}}_n^{(l)}\}$. It's required that $\widehat{\boldsymbol{\beta}}_{org}$ can update adaptively, according to the new observation set, i.e., $\widetilde{O} = \bigcup_{i=1}^{l}\widetilde{\boldsymbol{o}}^{(l)}$. This requirement is two-fold: first, if new observations show gradual change of relation between dependent and independent variables (resulting from non-stationarity of observations), the current model should update to accommodate the change accordingly; second, if not, the accuracy of the current model can be improved by using new observations despite outliers.

However adaptive updating is nontrivial. First, it is very hard to distinguish outliers from those normal ones which impel the change of dependency between variables. Second, even the non-stationarity of observations can be verified, it's very hard to tell if the change is temporary or long-term, thus we should neither completely accept nor ignore the new dependency. Furthermore, out-of-date observations in the current model which no longer accommodate to new dependency should be removed from new model, yet deciding those out-of-date observations is also not easy.

To update $\widehat{\boldsymbol{\beta}}_{org}$ adaptively by utilizing new observations, we apply a two step updating scheme, where 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. We explain our design in detail in the following subsections.

3.3.1 Calculating New Rough Model

In this procedure, a new 'safe' subset of $\widetilde{O}$ is formed and included into $\widehat{\boldsymbol{\beta}}_{org}$ to construct the new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$. The advantage is that half new observations will be included into the new rough model, such that the model will reflect new dependency to some extent, while minimizing the risk of bringing outliers into the model.

Specifically, S cooperates with each $\widetilde{N}_i$ to select the new optimal 'safe' subset (with a size of $\lceil\frac{n}{2}\rceil$), following the protocols which have been introduced in Section 3.2.1. We denote the new optimal local 'safe' observation subset of $\widetilde{N}_l$ as $\widetilde{\boldsymbol{o}}^{(c_l)}$. Each $\widetilde{N}_l$ submits $(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{x}}^{(c_l)}$ and $(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{y}}^{(c_l)}$with masking to S. After S obtaining the aggregates, i.e., $\sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{x}}^{(c_l)}$ and $\sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{y}}^{(c_l)}$, it estimates a new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$ by using

$$\begin{aligned}
\widetilde{\boldsymbol{u}}^{(rgh)} &= \boldsymbol{u}^{(org)} + \sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{x}}^{(c_l)},\\
\widetilde{\boldsymbol{v}}^{(rgh)} &= \boldsymbol{v}^{(org)} + \sum_{l=1}^{q}(\widetilde{\boldsymbol{x}}^{(c_l)})^T\widetilde{\boldsymbol{y}}^{(c_l)},\\
\widetilde{\boldsymbol{\beta}}_{rgh} &= (\widetilde{\boldsymbol{u}}^{(rgh)})^{-1}\widetilde{\boldsymbol{v}}^{(rgh)}.
\end{aligned}$$

3.3.2 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 old observations which have been included in $\widetilde{\boldsymbol{\beta}}_{rgh}$ will be rechecked, and those out-of-date ones will be removed, while new observations which haven't been embedded in $\widetilde{\boldsymbol{\beta}}_{rgh}$, i.e., 'uncertain' ones, will be tested such that non-outliers fitting $\widetilde{\boldsymbol{\beta}}_{rgh}$ well will be added into the model. In this way, a new refined model is derived.

In detail, S communicates with each participant in $\mathbb{U}$, to test the outlyingness of observations in $\boldsymbol{o}^{(c_l)} \bigcup \widetilde{\boldsymbol{o}}^{(r_l)}$, in accordance to $\widetilde{\boldsymbol{\beta}}_{rgh}$, which is alike the procedure obtaining $\boldsymbol{o}^{(f_i)}$ (described in Section 3.2.2). Those observations in $\boldsymbol{o}^{(c_l)}$ which unfit $\widetilde{\boldsymbol{\beta}}_{rgh}$ will be removed from the new model. On the other side, observations in $\widetilde{\boldsymbol{o}}^{(r_l)}$ which pass the test will be added into the new model.

The advantage of the design is also two-fold. First, once an observation is marked as out-of-date, it will be removed, and never be brought back into the model again. It means that as the model is updated, out-of-date observations will be excluded from latest model gradually, and only those observations which always accord with the relation between variables can survive. Second, fresh observations can be included into the new model as much as possible for improving model accuracy, at the same time, those fresh ones which fail to fit into the model (implying outliers) will be ignored.

3.4 Secure Aggregation and Comparison

3.4.1 One-Time Pad Masking

Remember that in Section 3.2.1, the summation of certain kind of local aggregated results (belonging to participants) should be calculated by S without disclosing individual ones. We propose a lightweight one-time pad masking

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

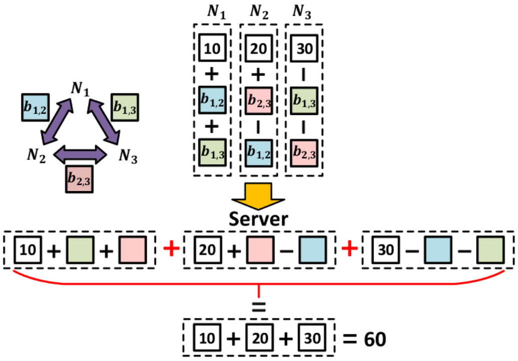


Fig. 5. In this example of one-time pad masking, the server calculates the summation of 10, 20, and 30 held by N_1, N_2 and N_3.

technique, and explain the key idea by calculating $\sum_{i=1}^{m} \boldsymbol{s}^{(i)}$ ($\boldsymbol{s}^{(i)}$ is hold by N_i) as follows:

The basic idea of masking is that each $\boldsymbol{s}^{(i)}$ is masked with certain secret value, such that all masks can be cancelled when they are added on the server side. Suppose each pair of participants (N_i, N_k) agrees on certain random value $b_{i,k}$. 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:

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

We give a three person example in Fig. 5. N_1 shares secret values $b_{1,2}$ and $b_{1,3}$ with N_2 and N_3, respectively, and N_2 and N_3 shares $b_{2,3}$. Each person has a private number, e.g, N_1 holds 10, and the server needs to calculated the summation of three private number, i.e., $10 + 20 + 30$. To this end, each person masks its private number with all secret values sharing with others before submitting it to the server , e.g., N_1 masks 10 with $b_{1,2}$ and $b_{1,2}$ (i.e., calculating $10 + b_{1,2} + b_{1,3}$), and then reports $10 + b_{1,2} + b_{1,3}$ to the server. Notice that N_2 subtracts $b_{1,2}$ from 20, since N_1 adds it on 10. After collecting all masked values from participants, the server calculates$(10 + b_{1,2} + b_{1,3}) + (20 - b_{1,2} + b_{2,3}) + (30 - b_{1,3} - b_{2,3}) = 10 + 20 + 30$.

In order to avoid exchanging random values $b_{i,k}$ 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 $s_{i,k}$, 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 $h(b_{i,k})$.

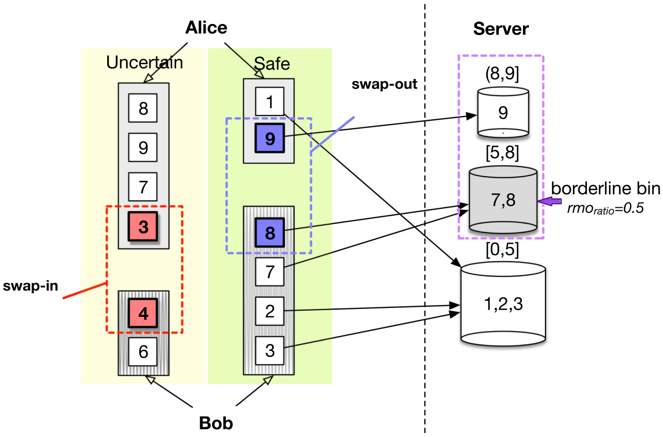


Fig. 6. In this example of blocking, observations with residuals 9 and 8 in the ‘safe’ subset are replaced with observations with residuals 3 and 4 in the ‘uncertain’ subset.

Remarks. In practice, it does not necessarily mask $\boldsymbol{s}^{(i)}$ using secrets shared with all others. Participants can be divided into groups, such that each participant only shares maskings with individuals belonging to the same group. Thus $\boldsymbol{s}^{(i)}$ is masked with secrets sharing with its group members.

3.4.2 Blocking

In the third step of observation-swapping protocol, it is necessary to identify in_{num} observations with maximum residuals in current global 'safe' subset, forming a swap-out subset. Thus S needs to sort all those observations according to residuals, and decides a swap-out threshold of residual. To this end, residuals of observations should be compared. instead of collecting and comparing residuals directly, which is vulnerable to observation recovery attacks, S runs a blocking-based secure comparison protocol under the cooperation of participants.

Specifically, S publishes a set of bins such that a residual can fall into one and only one bin, furthermore, the sizes 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 current 'safe' subset 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, in which observations in smaller bins well be removed first. S decides the borderline bin bin_{bd} such that all observations in bins smaller and bigger than it will be removed and kept, respectively. While partial observations in bin_{bd} will be removed. Then S broadcasts the bin_{bd} and removed ratio of observations rmo_{ratio} in it. According to this, N_i selects all observations in bins smaller than bin_{bd}, and randomly selects observations in bin_{bd} with a probability of rmo_{ratio} to form local swap-out subset $\boldsymbol{o}^{(out_i)}$.

We give an example in Fig. 6. Both Alice and Bob have theirs local ‘safe’ subsets, the residuals of which are $\{1, 9\}$ and $\{8, 7, 2, 3\}$, respectively. Alice and Bob decide their local swap-in subsets (i.e., observations with residuals 3 and 4, selecting from their local ‘uncertain’ subsets, the residuals of which are $\{8, 9, 7, 3\}$ and $\{4, 6\}$, respectively), according to the MRE announced by the server, and let the server know the total number of observations in swap-in subset, i.e., 2. Then the server publishes three bins of residuals, i.e., $[0, 5), [5, 8], (8, 9]$. Alice and Bob check their own 'safe' subsets, and repot the total numbers of observations falling in each bin, i.e., *three*, *two* and *one* observations fall into $[0, 5), [5, 8]$ and (8,9]. Since the server knows that two observations should be swapped out from the global 'safe' subset, it decides that the borderline bin is [5,8], which means that the observation in bin (8,9] will be swapped out. Furthermore, the server sets $rom_{ratio} = 0.5$, i.e., one swap-out observation in bin [5,8] will be selected randomly. Consequently, Alice knows observation with residual *9* will be swapped out, and Bob chooses to swap the observation with residual *8* out.

4 Analysis

4.1 Computational Complexity on Mobile Devices

In Lotus, the main computations are matrix multiplications, thus we use the number of *multiplications* and *additions* to represent the computational complexity. Since, the cost on initial model estimating and model evloving is exactly the same, we take the initial model estimating as an example to present the detailed analysis as follows:

4.1.1 Selecting Primary 'Safe' Subset

In order to pick up a primary 'safe' subset from local observations, N_i needs to calculate covariance matrix $\boldsymbol{V}^{(i)}$ and d_M, which takes $n(p+1)^2$ and $n(p+1)(p+2)$ multiplications, $(n-1)(p+1)(p+3)$ and $np(p+2)$ additions, respectively. Then N_i calculates $(\boldsymbol{x}^{(c_i)})^T(\boldsymbol{x}^{(c_i)})$ and $(\boldsymbol{x}^{(c_i)})^T\boldsymbol{y}^{(c_i)}$ based on its local primary 'safe' subset. In the case that all local observations of N_i are considered as 'safe', which means the set of all those observations, i.e., o_i, will be used to estimate a primary rough model acting as the start point of the hill-climbing algorithm, $n(p+1)(p+2)$ multiplications and $(n-1)(p+1)(p+2)$ additions should be conducted for calculating $(\boldsymbol{x}_j^{(i)})^T\boldsymbol{x}_j^{(i)}$ and $(\boldsymbol{x}_j^{(i)})^T\boldsymbol{y}_j^{(i)}$ in total. In summary, the complexity of multiplication and addition operation of selecting primary 'safe' subset is $O(np^2)$ and $O(np)$, respectively.

4.1.2 Approaching Optimal 'Safe' Subset

Since $(\boldsymbol{x}_j^{(i)})^T\boldsymbol{x}_j^{(i)}$ and $(\boldsymbol{x}_j^{(i)})^T\boldsymbol{y}_j^{(i)}$ have been calculated in the last step, N_i only computes n and $n(p+1)$ multiplications, and $n-1$ and $n(p+1)$ additions for calculating residuals of its all local observations and $(\boldsymbol{y}^{(c_i)})^T\boldsymbol{y}^{(c_i)}$ in each iterations, respectively. In summary, the complexity of multiplication and addition operation of approaching optimal 'safe' subset is $O(np)$ and $O(np)$, respectively.

4.1.3 Deriving Refined Global Model

For checking the outlyingness of local observations in the remaining subset of N_i, it calculates the residuals of those observations, i.e., $\boldsymbol{e}^{(i)}$ and $z_j^{(i)}$, which takes $n(p+2)$ multiplications and n additions, in total. Consequently, the complexity of multiplication and addition operation of deriving refined global model is $O(np)$ and $O(n)$, respectively.

4.2 Security Analysis

4.2.1 Observation Recovery Attacks

After deciding a primary 'safe' subset $\boldsymbol{o}^{(\boldsymbol{c})}$, S estimates $\widehat{\boldsymbol{\beta}}_0$ based on it. Not surprisingly, S can build a set of equations with unknowns of $\boldsymbol{o}^{(\boldsymbol{c})}$, thus we introduce the following theorem:

Theorem 2. *In order to defend against a primary 'safe' subset* $\boldsymbol{o}^{(\boldsymbol{c})}$ *(i.e.,* $\bigcup_{i=1}^{m}\boldsymbol{o}^{(c_i)}$*) from been recovered by* S *or other attackers, the number of observations collected by all participants, i.e., the size of* $\mathbf{O}$*, 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, $\boldsymbol{o}^{(\boldsymbol{c})}$ has at least $p+1$ observations in order to estimate a regression. Thus, the following inequations hold,

$$\lceil m\times n/2\rceil \times (p+1) \;>\; (p+1)+(p+1)(p+2)/2 \tag{5}$$

$$\lceil m\times n/2\rceil \geq p+1. \tag{6}$$

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

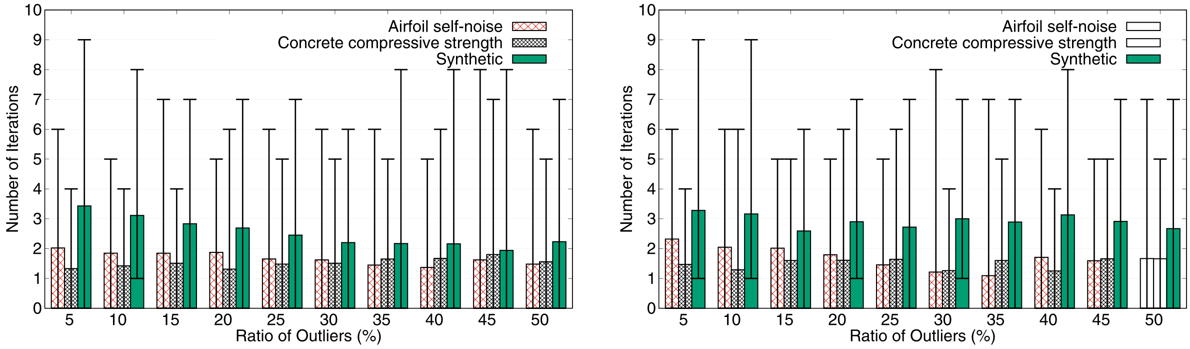
In this step, S obtains $\boldsymbol{\mu}$ and $\boldsymbol{V}$ of all $m\times n$ observations from all participants, which implies that S can build $\frac{(p+1)(p+2)}{2}-1$ and $p+1$ equations related to $\boldsymbol{O}$, according to them, respectively. Note that the size of $\boldsymbol{O}$ is $m\times n\geq 2p+3$ (according to Theorem 2). Therefore, the number of variables is at least $(2p+3)(p+1)$. Since $(2p+3)(p+1) > \frac{(p+1)(p+2)}{2} + p$ always holds, S can not build enough equations to recover $\boldsymbol{O}$.

Additionally, original observations can be protected from being recovered during approaching optimal 'safe' subset and deriving refined global model, since observations swapped in or out of 'safe' subset could be viewed as adding a matrix onto a matrix, or subtracting a matrix from it, it is impossible for S to deduce any information by the differences between two received matrices.

4.2.2 Collusion Attacks

Malicious participants may collude to recover local aggregated values of N_i, which can be used to further deduce the related observations. In the example of calculating $\sum_{i=1}^{m}\boldsymbol{s}^{(i)}$ in Section 3.4.1, $\boldsymbol{s}^{(i)}$ of N_i is masked with $m-1$ one-time pads sharing with $m-1$ different participants. That is to say, in order to recover $\boldsymbol{s}^{(i)}$, all the $m-1$ participants are malicious and collude with each other. Given M colluding participants, there are two cases: first, if $M \geqslant m-1$, then the probability that all participants sharing masks with N_i collude is $\mathbb{P}(M) = (\frac{M}{q})^{(m-1)}$. As $M \ll q$, $\mathbb{P}(M)$ is extremely small; second, if $M < m-1$, it's impossible to recover $\boldsymbol{s}^{(i)}$.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.



(a) Random Noise

(b) Normal Noise

Fig. 7. The number of iterations needed for optimizing ’safe’ subset versus outlier ratio.

5 Performance Evaluation

5.1 Methodology

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

(1) *Airfoil self-noise* [16]: 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* [18]: 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 μ and a standard deviation of σ, where μ and σ 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 estimate 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 ith diagonal element refers to the influence of the ith 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 *acc* of an estimate by calculating the relative difference of model coefficients between an estimated $\widehat{\boldsymbol{\beta}}$ and the ground truth $\boldsymbol{\beta}_*$, i.e.,

$$acc = \frac{\|\boldsymbol{\beta}_* - \widehat{\boldsymbol{\beta}}\|}{\|\boldsymbol{\beta}_*\|}.$$

For each dataset, noise type and each simulation configuration, we run the simulation 20 times and get the average.

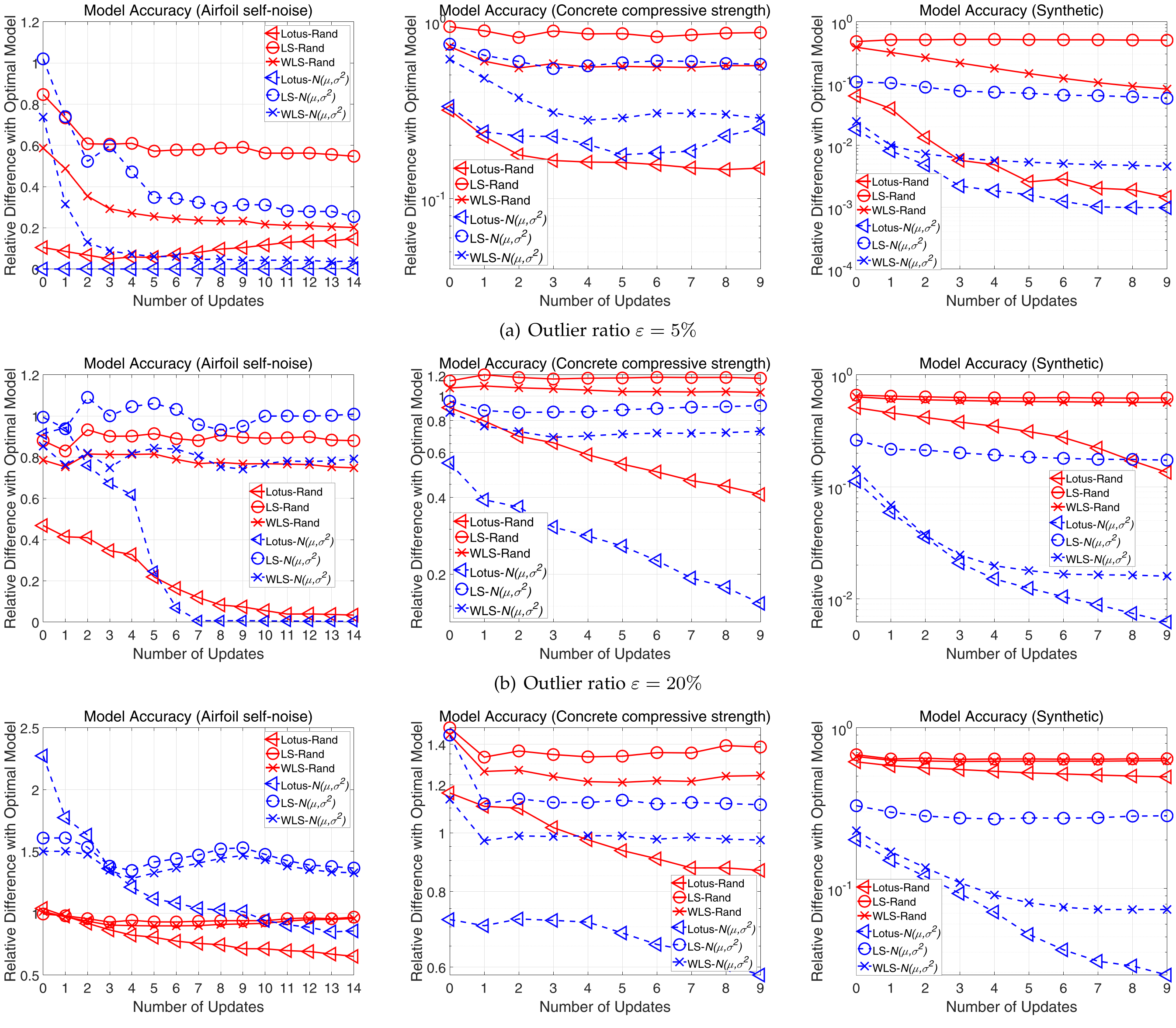
5.2 Number of Iterations Needed for Rough Global Model Estimating

As a larger number of iterations mean more interactions between participants and the server, resulting in a larger communication cost in modeling and larger privacy risk, we examine how many iterations are needed to optimize a ‘safe’ subset. We use above datasets in normally distributed and random noise settings. We vary the ratio of outlier from 5 to 50 percent at an interval of 5 percent and run the experiments 20 times and get the average number of iterations. We have following observations according to experimental results shown in Fig. 7. First, for two real datasets, the average number of iterations in all noise settings is no more than 2 (expect for Airfoil dataset with 5 percent outliers). Second, for synthetic dataset, the maximum average number of iterations is 3.4, and the typical average number is around 3. Third, comparing with real data, synthetic data need more iterations. We speculate that it is because the linear relationship between synthetic observations is stronger than that of real data, which implies more borderline observations. Fourth, the number of iterations decreases slightly with the increasing of outlier ratio. We find that residual differences between normal data are smaller than that between normal data and outliers, thus more effort is needed for optimizing ‘safe’ subset. Finally, the number of iterations has no obvious difference under different types of noise, especially for real datasets.

5.3 Accuracy with Model Updating

In this experiment, we examine the performance of Lotus on the three 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 [19]. For *Airfoil self-noise*, *Concrete compressive strength* and

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.



(a) Outlier ratio $\varepsilon = 5\%$

(b) Outlier ratio $\varepsilon = 20\%$

(c) Outlier ratio $\varepsilon = 40\%$

Fig. 8. Model accuracy versus the number of updates under different outlier ratios.

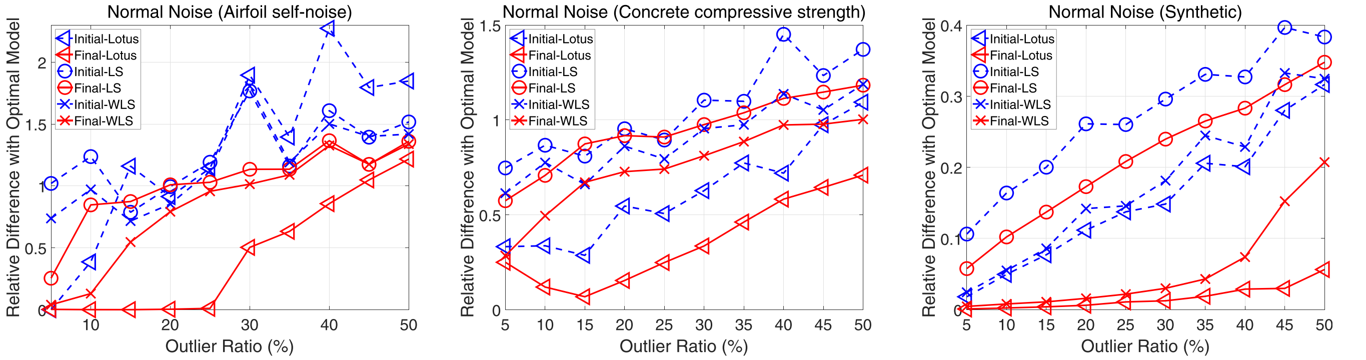
Synthetic, the number of groups are 15, 6, 10 and 10, respectively. We then randomly pick up one unused group from one certain dataset for model estimating or updating. We compare Lotus with LS and WLS, under both the random and normal distributed noise settings and three typical outlier ratios, i.e., 5, 20 and 40 percent, implying high, medium and low quality data, and we illustrate the experimental results in Figs. 8a, 8b, and 8c, respectively.

Fig. 8 plots the accuracy of estimated models, where ground truths are 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 datasets. Although, in few settings, e.g., Airfoil data with 40 percent outliers, WLS can achieve 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.

5.4 Robustness under Different Outlier Ratios

In this experiment, we further examine the robustness of Lotus against outliers under different outlier ratios ε, 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 ε varies from 5 to 50 percent at an interval of 5 percent. 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 randomly distribute them to observations in the group.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.



(a) Airfoil self-noise

(b) Concrete compressive strength

(c) Synthetic

Fig. 9. Model accuracy versus outlier ratio (under $N(\mu, \sigma^2)$ normal noise setting).

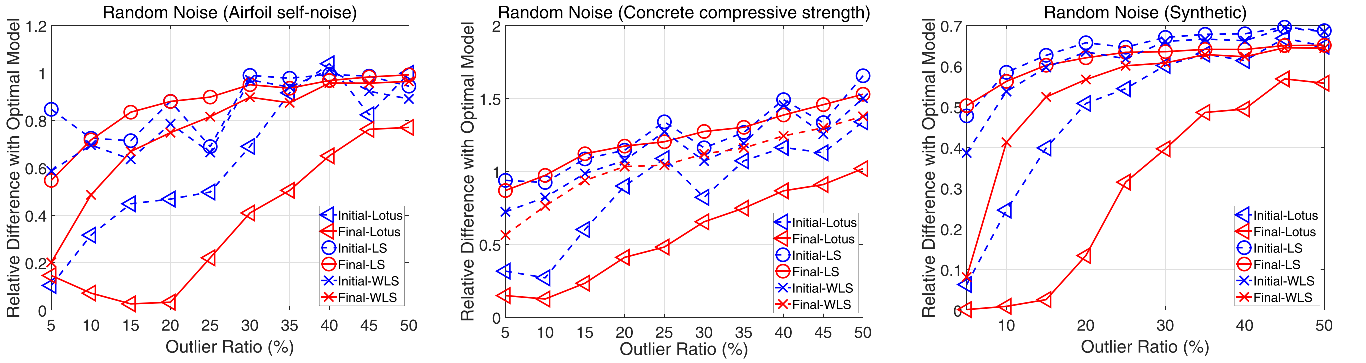
Figs. 9 and 10 plot the accuracy of estimated models (ground truth is the corresponding global optimal models) as a function of outlier ratio under random and normal distribution noise settings, respectively. It can be seen that Lotus outwits LS and WLS in all settings. LS is very sensitive to outliers, even under a low outlier ratio of 5 percent, LS gets a bad performance, whereas, Lotus can hold good accuracy even when ε increases to 40 percent, 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 percent. 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.

6 Prototype Implementation

We implement Lotus on Google Nexus 5X smartphones (each of which is equipped with a hexa-core 1.8 GHz CPU and 2 GB memory, running on Android 8.0), Huawei Honor 5X (KIW-UL00) smartphones (eight-core 1.5 GHz and 2 GB memory, running on EMUI 4.0.3, based on Android 6.0), and a server (with Inter I7 3.5 GHz six-core CPU and 32 GB RAM, running on Windows 10). We use smartphones to serve as different participants in a participatory sensing task, and set the server to collect sensory data from participants and to coordinate the regression modeling procedure accordingly. Data communication between participants and the server is based on WLAN in our laboratory. With this prototype, we conduct experiments based on a dataset of *Energy efficiency* [17], which consists of 763 observations. We use attributes of *relative compactness*, *orientation*, *roof area*, *overall height*, *glazing area*, and *glazing area distribution* to build the influence function of *heating load* (p is 6).

6.1 Performance on Model Accuracy

In this experiment, we randomly divide all observations into 7 groups, each of which contains 109 observations, which satisfies that at least $n \geq 50 + 8p$ observations are included in each group according to [19], and randomly pick up one unused group for estimating an original model or updating the model. Then, we further divide each observation group into 8 subgroups (each contains 13 or 14 observations) and distribute them to all participants randomly. For each observation group, outliers are generated by adding certain type of noise on observations. Specifically, we apply three types of noise, i.e., *random noise* and $N(\mu, \sigma^2)$ described in Section 5.1, and *standard normal distributed noise satisfying* $N(0, 1)$. For each type of noise, we vary the outlier ratio ε from 5 to 50 percent at an interval of 5 percent. Under each setting of noise (outlier ratio ε and types of noise), the server generates $109 \cdot \varepsilon$ tokens and randomly distributes them to all participants (g.e., for $\varepsilon = 10\%$, 11 tokens are generated). For each token, a participant randomly picks one observation and produces an outlier observation by superposing the noise (generated according to the certain noise type) on the observation. For each noise setting and outlier ratio, server performs regression modeling adopting Lotus, LS and WLS, respectively. We evaluate model accuracy by conducting each experiment 20 times and calculating the average.



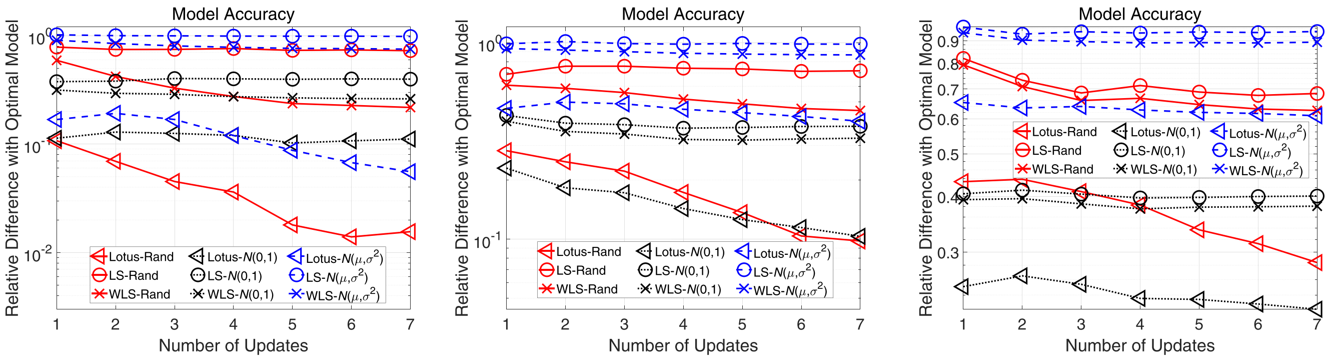
(a) Airfoil self-noise

(b) Concrete compressive strength

(c) Synthetic

Fig. 10. Model accuracy versus outlier ratio (under random noise setting).

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.



(a) Outlier ratio $\varepsilon = 10\%$ (b) Outlier ratio $\varepsilon = 20\%$ (c) Outlier ratio $\varepsilon = 30\%$

Fig. 11. Model accuracy versus the number of updates under different outlier ratios.

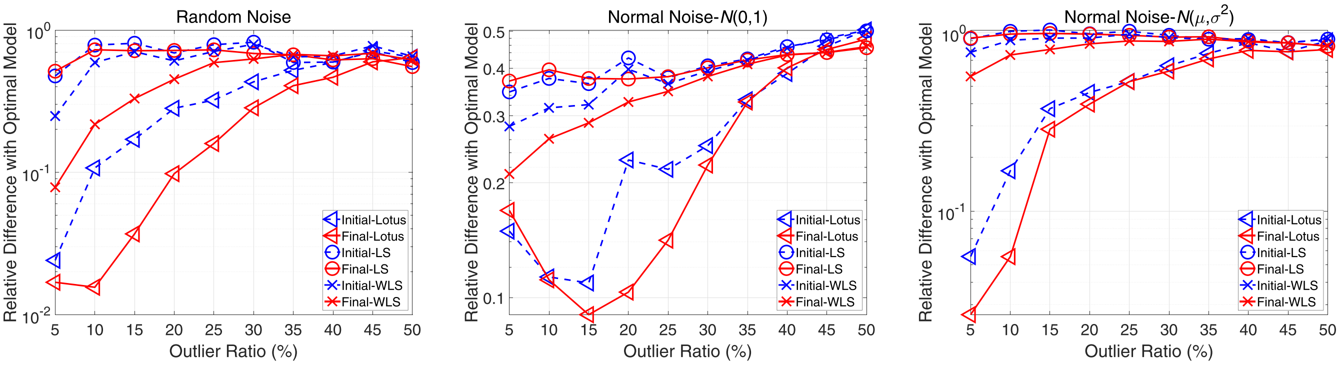
In order to examine the performance of Lotus with model updating. We calculate average relative differences of model coefficients between estimated models and global optimal models (estimated using the same way as described in Section 5.3). Figs. 11a, 11b, and 11c plot average relative differences as a function of the number of updating under 10, 20 and 30 percent outliers, respectively. It can be seen that accuracy of models estimated using LS and WLS remains unchanged in despite of updating. In contrast, accuracy of Lotus improves along with updating in most cases. Especially for random noise outliers, model updating help improving model accuracy significantly, under all three ratios of outliers. With $N(\mu, \sigma^2)$ distributed noise, updating gains less improvement of accuracy as the increasing of outlier ratio, which is because $N(\mu, \sigma^2)$ noise leads up to larger errors of observations comparing with other two types of noise.

We also verify the robustness of Lotus against outliers under different outlier ratios, and of different types. Fig. 12 plots the average model accuracy as a function of outlier ratio under different noise settings. Both initial estimates and final models are illustrated. According to Fig. 12, we make following conclusions: First, Lotus can achieve excellent accuracy when percentage of standard normal and randomly distributed noise is less than 20 percent, and percentage of $N(\mu, \sigma^2)$ distributed noise is less than 10 percent. Second, Lotus outwits LS and WLS in all settings, especially when percentage of outlier is less than 40 percent. For instance, when percentage of outlier is equals to 10 percent, the final estimate accuracy of Lotus can be one order of magnitude higher than LS and WLS. Third, the accuracy decrease of Lotus is the slowest as the increase of outlier ratio. Overall, for Lotus, a final estimate is more accurate than its corresponding initial estimate under low and moderate outlier ratios. The accuracy difference between final and initial models reduces, because too many outliers make the choice of optimal 'safe' subset during updating much harder, which impedes the refinement of estimate.

Additionally, we measure iterations needed for approaching optimal 'safe' subset under three types of noise. we vary outlier ratio from 5 to 50 percent at an interval of 5 percent and run the experiments 20 times and get the average number of iterations. Experimental result is depicted in Fig. 13, which demonstrates that the average number of iterations are no more than 3 under any type of noise. Furthermore, we can conclude that the average numbers of iterations are also robust to the percentage of outliers. Finally, we find that random noise outliers imply more iterations. We consider that LS-based estimators make an assumption that gross errors satisfy normal distribution, thus random noise will lead to larger deviation of initial 'safe' subset from the optimal one, and consequently, more iterations are needed.

6.2 Computational Complexity

In this experiment, we aim to measure time consumption of executing Lotus on mobile devices. We also use the observation groups of *Energy efficiency*, each of which contains 109 observations. We vary the number of participants involved in regression, denoted as m, from 3 (the minimum number of participants should be included in one-time pad masking based aggregations) to 15 (we assume that at least $p+1$, i.e., 7 observations should be collected by each participants),



(a) Random noise (b) Normal noise *N(0,1)* (c) Normal noise $N(\mu, \sigma^2)$

Fig. 12. Model accuracy versus outlier ratio under different noise settings.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

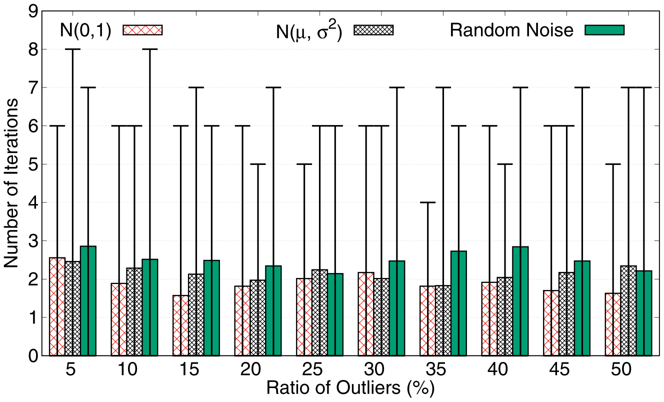


Fig. 13. Number of iterations needed versus outlier ratio under different noise settings.

and calculate the number of observations obtained by each participants, i.e., $n = \frac{109}{m}$. We measure the time consumption on steps of *selecting primary 'safe' subset*, *approaching optimal 'safe' subset*, and *deriving refined global model* under different numbers of observations. Figs. 14a and 14b are the boxplots of time consumption on Huawei Honor and Google Nexus smartphones, respectively. It can be seen that time consumption increases linearly with the number of observations n in all steps (notice we skip several impossible values of n, e.g., 14, 16, 17, 28-35, etc.). Surprisingly, we find that, although the hardware configurations of Huawei Honor 5X and Google Nexus 5X smartphones are comparable, the time consumption on Nexus is one order of magnitude lower than that on Huawei. We speculate it may because Nexus runs on a higher version of Android.

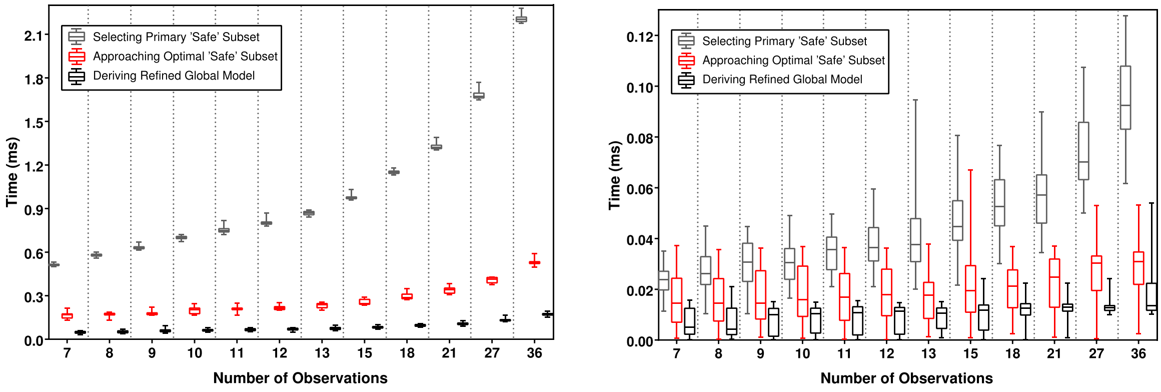
7 Related Work

Privacy-preserving data aggregation has 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 [20] 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.

Some researchers focus on robust regression without considering privacy issue. For example, M. Jagielski et al., investigate poisoning attacks on regression learning, and propose a defense approach TRIM, which iteratively estimates the regression parameters based on a subset of points with lowest residuals in each iteration [21]. S. Lathuiliere et al. derive an optimization algorithm, DeepGUM, which can adapt to the updating of outlier distribution. DeepGUM requires a supervised training with clean samples [22]. However, distinguishing clean samples from outliers is non-trivial. 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.

8 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



(a) Huawei Honor 5X

(b) Google Nexus 5X

Fig. 14. Computation time on smartphones versus the number of observations.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.

suspected outliers and withdrawing outdated observations during model updating. Both analysis and extensive simulation results demonstrate the efficacy of Lotus. We have also built a prototype system of MCS in our lab, and further examined the feasibility of Lotus under real deployment.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61672151, 61772340, 61420106010), Shanghai Rising-Star Program (Grant No.17QA1400100), National Key R&D Program of China (Grant No. 2018YFC1900700), Shanghai Municipal Natural Science Foundation (Grant No. 18ZR1401200), the Fundamental Research Funds for the Central Universities (Grant No. EG2018028), DHU Distinguished Young Professor Program, and 2017 CCF-IFAA Research Fund.

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," *IFAC-Papers OnLine*, vol. 49, 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," *Environmental Sci. Technol.*, vol. 50, pp. 6546–6555, 2016.

[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 31st Int. Conf. Advanced Inf. Netw. Appl.*, 2017, pp. 1042–1049.

[4] Y. Gong, Y. Fang, and Y. Guo, "Private data analytics on biomedical sensing data via distributed computation," *IEEE/ACM Trans. Comput. Biology Bioinf.*, vol. 13, no. 3, pp. 431–444, May-Jun 2015.

[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," *Comput. Graphical Statist.*, vol. 14, pp. 263–279, 2004.

[7] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Netw.*, vol. 11, pp. 1062–1074, 2013.

[8] A. D. Paola, S. Gaglio, G. L. Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Trans. Cybernetics*, vol. 45, no. 5, pp. 902–913, May 2015.

[9] J. Vaidya and C. Clifton, "Privacy-preserving outlier detection," in *Proc. IEEE 9th Int. Conf. Young Comput. Sci.*, 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," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1199–1211, Apr. 2016.

[11] M. Kutner, C. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*, New York, NY, USA: McGraw-Hill, 2005.

[12] D. Ruppert, and M. P. Wand, "Multivariate locally weighted least squares regression," *Ann. Statist.*, vol. 22, pp. 1346–1370, 1994.

[13] P. Mahalanobis, "On the generalised distance in statistics," *Proc. Nat. Inst. Sci.*, vol. 2, pp. 49–55, 1936.

[14] R. Hemmecke, M. Köppe, J. Lee, and R. Weismantel, "Nonlinear integer programming," in *50 Years of Integer Programming 1958–2008*, Berlin, Germany: Springer, 2009, pp. 561-618.

[15] M. R. Garey, and D. S. Johnson, *Computers and Intractability a Guide to the theory of NP-Completeness*, San Francisco, CA, USA: Freeman, 1979.

[16] The Airfoil self-noise dataset, [Online]. Available: [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Airfoil+Self-Noise, Accessed: Mar. 4, 2014.

[17] Energy efficiency dataset, [Online]. Availiable: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency, Accessed: Nov. 30, 2012.

[18] Concrete compressive strength dataset, [Online]. Availiable: https://archive.ics.uci.edu/ml/datasets/concrete+compressive+strength, Accessed: Aug. 3, 2007.

[19] S. Green, "How many subjects does it take to do a regression analysis?," *Multivariate Behavioral Res.*, vol. 26, pp. 499–510, 1991.

[20] R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher, "PoolView: Stream privacy for grassorts participatory sensing," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 281–294.

[21] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. N-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Security Privacy*, 2018, pp. 19–35.

[22] S. Lathuiliere, P. Mesejo, X. A-Pineda, and R. Horaud, "DeepGUM: Learning deep robust regression with a gaussian-uniform mixture model," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 202–217.

Shan Chang received the PhD degree in computer software and theory from Xián Jiaotong University, in 2013. From 2009 to 2010, she was a visiting scholar with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. She was also a visiting scholar with the BBCR Research Lab, University of Waterloo, from 2010 to 2011. She is now an associate professor with the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and sensor networks. She is a member of the IEEE.

Chao Li received the BS degree in computer science and technology from Anhui University, in 2016. He is currently working toward the postgraduate degree in the Department of Computer Science and Technology, Donghua University, Shanghai. His research interests include privacy and distributed system security. He is a member of the ACM.

Hongzi Zhu received the PhD degree in computer science from Shanghai Jiao Tong University, in 2009. He was a post-doctoral fellow with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and the Department of Electrical and Computer Engineering, University of Waterloo, in 2009 and 2010, respectively. He is now an associate professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include vehicular networks, mobile sensing, and computing. He received the Best Paper Award from IEEE Globecom 2016. He was a leading guest editor for *IEEE Network Magazine*. He is an associate editor for the *IEEE Transactions on Vehicular Technology*. He is a member of the IEEE, IEEE Computer Society, IEEE Communication Society, and IEEE Vehicular Technology Society. For more information, please visit http://lion.sjtu.edu.cn.

Hang Chen received the BS degree from the Shanghai University of Engineering and Science, in 2017. He is currently working toward the postgraduate degree in the Department of Computer Science and Technology, Donghua University, Shanghai. His research interests include Internet of Things, mobile sensing, and wearable computing. He is a member of the ACM.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: Shanghai Jiaotong University. Downloaded on October 26,2020 at 06:17:01 UTC from IEEE Xplore. Restrictions apply.