Sieve: Lightweight Robust Regression on Private Sensory Data

Shan Chang*, Hang Chen*, Chao Li* Hongzi Zhu†, 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
Email: changshan@dhu.edu.cn, {chenhang, chaoli}@mail.dhu.edu.cn, hongzi@cs.sjtu.edu.cn, luting@dhu.edu.cn

***Abstract*—Mobile crowd sensing (MCS) data have unique features, i.e., private, error-prone, non-stationary and opportunistically generated, and collected by resource-constrained mobile devices, which bring the regression task new challenges. First, it is of great difficulty to derive an accurate model without acquiring raw data. Second, adaptive model updating mechanism is urgently needed. Last, regression schemes should be lightweight. In this paper, we propose a blind regression scheme, called Sieve, in MCS settings. The core idea is first to let the server help in coordinating the selection of a small 'clean' subset of observations locally stored over all volunteers. Based on the 'clean' subset, an initial model can be established among volunteers in a distributed fashion. With observations constantly coming, instead of model re-estimating from the scratch, newly selected 'clean' observations are used to update the established model. Sieve preserves data privacy by only exchanging aggregated information. With the incremental model updating strategy, it also minimizes the communication and computation overhead of mobile devices. Extensive trace-driven simulations are conducted and the results demonstrate the efficacy of the Sieve design.**

***Index Terms*—private sensory data; robust regression; adaptive model updating; outlier**

I. Introduction

The paradigm of mobile crowd sensing (MCS) is developed, as mobile devices equipped with various sensors having become ubiquitous. In MCS, a large group of mobile device owners report sensory data, i.e., observations, to a server, and the server measures, estimates or infers phenomena and processes of common interest based on them, which facilitates a larger number of new applications, such as, smart home controlling, air pollution estimation, and health model forecasting etc.

Regression is a powerful statistical machine learning method, which estimates unknown parameters of a given model based on collected observations. MCS application scenarios pose four rigid requirements to a practical regression estimator as follows. *1) Privacy friendly:* as observations collected by mobile devices are private, raw observations should keep locally stored and processed during regression estimation. *2) Robust to outliers:* untrained volunteers are recruited in sensing task with low-end mobile devices equipped with non-dedicated sensors, which implies that untrustworthy low-quality observations or outliers are prevalent. Hence, regression estimator should be able to tolerate high ratio of low-quality data. *3) Adaptive model updating mechanism:* observations are naturally non-stationary, and are collected in an opportunistic fashion. It implies that the distribution of observations might change over time. The estimator should be able to deal with ever-coming and ever-changing observations and update models adaptively to current trend of observations. *4) Ultralow overheads:* mobile devices 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 this paper, we propose a regression estimator for MCS applications, called *Sieve*, implemented as a set of protocols running on the MCS server and mobile devices of volunteers. To deal with opportunistic and non-stationary observations, as illustrated in Figure 1, Sieve first estimates an initial model and periodically updates the model with newly collected observations. More specifically, to estimate the initial model, a small 'clean' (not-likely-to-be-outlier) subset of observations distributed over all volunteers is first determined. Then, an aggregation of such observations is calculated locally at each volunteer and then mixed with the counterparts at other volunteers before being sent to the server to estimate a rough global model. After that, by examining the validity of local observations with this rough estimate, more 'clean' observations can be identified and utilized to refine the global estimate in a similar way. To update the model, a new 'clean' subset of observations is first determined from the new observations in the same way as adopted in estimating the initial model. Similarly, new mixed local aggregations of such 'clean' observations are received at the server, with which the server updates the initial global model and derives a new rough model. With the new rough model, each volunteer re-examines the validity of new observations and those observations that have already been involved in the initial model as well. The checking results in the form of aggregated values are reported to the server to build a new refined global model. We evaluate the performance of Sieve through extensive trace-driven simulations using both real and synthetic datasets. The results demonstrate the effectiveness and robustness of Sieve.

One main challenge in the design is to achieve reliable estimates on contaminated observations without revealing private observations, since it is hard to identify and remove outliers. Sieve tackles this challenge through two key strategies: 1) volunteers in Sieve not only take part in collecting observations but also collaborate with the server in making decisions. 2) Only aggregated results, which is proven secure, are exchanged between volunteers and provided to the server. Another challenge is to eliminate the impact of those out-of-date observations from the new model and to affect the new model with new effective observations in an incremental way. In Sieve, new observations will be selected and merged into the

978-1-5386-7646-2/19/$31.00 ©2019 IEEE

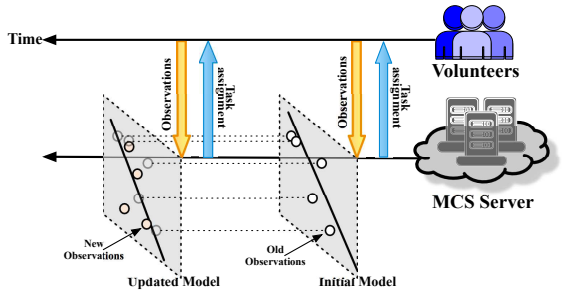

Fig. 1. Illustration of Sieve used in mobile crowd sensing applications.

current model, while old ones will be re-estimated according to new model. Aggregations of local observations leverage the additive property of multiplication of partitioned matrices. As a result, adding observations into or removing them from a model can be done in an incremental way.

II. Problem Formulation and Preliminaries

A. System Model

We consider typical MCS scenarios where sensory data are collected with mobile devices of unprofessional volunteers. Volunteers can communicate with each other and with the server via WiFi and 3G/4G. The features of volunteers and server are described as follows.

- **The server:** is greedy and rational. The purpose of the server is to achieve precise regression estimation and to update the model as needed. The server also wishes to pry private data of volunteers as much as possible.
- **Volunteers:** might be curious about the private data of others and try to get such information from any available intermediate results. They may also mutual collude (or with the server) to share knowledge in order to reveal more private information. However, we assume that the number of volunteers in collusion is limited.

B. Multivariate Linear Regression

We consider a set of volunteers $\mathbb{N} = \{N_1, \ldots, N_m\}$ and an application sever S. Each volunteer $N_i (0 \leqslant i \leqslant m)$ collects a set of its own observations. The j-th observation $\boldsymbol{o}_j^{(i)}$ of N_i can be represented as $[x_{j,1}^{(i)}, \ldots, x_{j,p}^{(i)}, y_j^{(i)}], (p > 1)$. S collects observations from volunteers and estimates a *multivariate linear regression (MLR) model* of observations as follows:

Definition 1. In MCS, an MLR model refers to observed independent variables $\boldsymbol{x}^{(i)} = [\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_{n_i}^{(i)}]^T$ (where $\boldsymbol{x}_j^{(i)} = [1, x_{j,1}^{(i)}, \ldots, x_{j,p}^{(i)}]$), and dependent variables $\boldsymbol{y}^{(i)} = [y_1^{(i)}, \ldots, y_{n_i}^{(i)}]^T$ from N_i, such that

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

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

According to the Least Square (LS) estimator, an estimate value of unknown $\boldsymbol{\beta}$, can be calculated as

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

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

C. Design Goals

- **Privacy Friendly**: raw observations cannot be obtained or inferred by any others during the regression.
- **Robust**: an observation is considered as a *regression outlier* if it deviates from the relation followed by the majority of the data. Robust refers that the derived relation still fits the majority of data even if the portion of regression outliers reaches up to 50% of all observations.
- **Incremental Updating:** in MCS, a regression model is updated periodically or once enough fresh observations have collected. Incremental model updating refers to a new model can be achieved according to the old regression coefficient vector $\boldsymbol{\beta}$, and a group of new observations, without re-estimating from scratch.
- **Adaptive:** model updating is adaptive if it can accommodate to the gradual changes of dependency between the predictor and criterion variables, resulting from the non-stationarity of observations.
- **Lightweight:** both regression and model updating should not induce too much communication and computational cost on mobile device (volunteer) side.

Remark. We do not restrict the outlier ratio from certain individual. However, given a group of observations for regression, the outliers ratio is limited up to 50%; otherwise, it is impossible to achieve a reasonable regression model, since 'bad' ones turn to be the majority of the data.

III. Design of Sieve

A. Overview

Sieve incorporates following two techniques, such that privacy preservation, model accuracy, and lightweight can be guaranteed, simultaneously.

Robust Regression Modeling. This technique includes three procedures. *1) Selecting minimum set of 'clean' observations: S* cooperates with volunteers to run a secure aggregation algorithm to help volunteers self-determine their own 'clean' observations. These observations form a global minimum 'clean' observation set (size $p + 2$), used for estimating a rough global model in the next step. *2) Building rough global model:* with own 'clean' observations, volunteers calculate corresponding aggregations locally, to contribute regression estimation. Before uploading an aggregated result to the server, a volunteer applies a slicing technique on it. *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 in a similar way as described above, resulting the ultimate initial global model.

Adaptive Model Updating. This technique includes two steps. *1) Calculating new rough model:* First, a new 'clean' subset is formed according to a fresh observation group, which follows the same procedure as in finding the initial 'clean' subset. Then the new 'clean' subset will be included into the current model securely and incrementally (by calculating proper aggregated results locally, and submitting them to S securely), resulting a new rough model. *2) Deriving new refined model:* based on the new rough model, each volunteer re-examines the quality of fresh observations as well as observations which have been included in the current model locally,

and decides which should be involved into or removed from the new model. Then, each volunteer calculates corresponding aggregation results and uploads them in a similar way as in step 1, enabling model updating based on the current model. Since none of raw data is revealed and there is no need to rebuild the model with all observations, privacy-preserving and lightweight updating can be achieved.

B. Robust Regression Modeling

1) Selecting Minimum Set of 'Clean' Observations: suppose N_i holds a set of observations $\boldsymbol{o}^{(i)} = \{\boldsymbol{o}_1^{(i)}, \ldots, \boldsymbol{o}_{n_i}^{(i)}\}$, where $\boldsymbol{o}_j^{(i)}$ indicates the *j*-th observation of N_i. Observations from a set of volunteers $\mathbb{N} = \{N_1, \ldots, N_m\}$ are denoted by $\boldsymbol{O} = \bigcup_{i=1}^{m} \boldsymbol{o}^{(i)}$. We choose $p+2$ observations from $\boldsymbol{O}$ with smallest Mahalanobis distance [5], denoted as d_M, to form a 'clean' subset which is presumably free of outliers. The reason is that d_M is commonly utilized to measure the deviation of a point from a distribution. A cutoff value of d_M for identifying outliers is recommended as $\sqrt{\chi^2_{(p,0.975)}}$ [6]. Given $\boldsymbol{o}_j^{(i)}$, from the observation set $\boldsymbol{O}$ with mean value $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{p+1}]$ and covariance matrix *V*, the corresponding Mahalanobis distance can be calculated as:

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

where covariance matrix V can be calculated as

$$\boldsymbol{V} = E\{(\boldsymbol{O} - \boldsymbol{\mu})^T (\boldsymbol{O} - \boldsymbol{\mu})\}. \tag{4}$$

On the purpose of protecting observations from being known by others, the owner of $\boldsymbol{o}_j^{(i)}$ calculates $d_M(\boldsymbol{o}_j^{(i)})$ locally, and S selects the minimum subset of 'clean' observations $\boldsymbol{O}_c$, by running **Algorithm 1** between $\mathbb{N}$ and S.

Algorithm 1: Selecting minimum set of 'clean' observations

1: /*Calculating $\boldsymbol{\mu}$*/.
2: N_i computes the sum of each column in $\boldsymbol{o}^{(i)}$ locally, i.e., $\boldsymbol{s}^{(i)} = (\sum_{j=1}^{n_i} x_{j,1}^{(i)}, \ldots, \sum_{j=1}^{n_i} x_{j,p}^{(i)}, \sum_{j=1}^{n_i} y_j^{(i)})$
3: N_i sends both $\boldsymbol{s}^{(i)}$ and n_i to S.
4: calculates $\boldsymbol{\mu} = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} \boldsymbol{s}^{(i)}$
5: S broadcasts $\boldsymbol{\mu}$ to $\mathbb{N}$.
6: /*Calculating $\boldsymbol{V}$*/.
7: N_i computes $\boldsymbol{V}^{(i)} = \sum_{j=1}^{n_i} (\boldsymbol{o}_j^{(i)} - \boldsymbol{\mu})^T (\boldsymbol{o}_j^{(i)} - \boldsymbol{\mu})$ and submits it to S.
8: S computes $\boldsymbol{V} = \frac{1}{\sum_{i=1}^{m} n_i} \sum_{i=1}^{m} \boldsymbol{V}^{(i)}$, and calculates $\boldsymbol{V}^{-1}$,
9: S broadcasts $\boldsymbol{V}^{-1}$ to $\mathbb{N}$.
10: /*Selecting the minimum set of 'clean' observations*/.
11: N_i computes $d_M(\boldsymbol{o}_j^{(i)})$ and sends the top-$(p+2)$ smallest $d_M(\boldsymbol{o}_j^{(i)})$ to S.
12: S sorts all $d_M(\boldsymbol{o}_j^{(i)})$ received, and picks top-$(p+2)$ smallest $d_M(\boldsymbol{o}_j^{(i)})$ to form $\boldsymbol{O}_c$.
13: S notifies the corresponding volunteers which observations have been selected as members of $\boldsymbol{O}_c$.

In Algorithm 1, N_i shares aggregated results rather than its raw observations $\boldsymbol{o}^{(i)}$. However, whether sharing those aggregated results will jeopardize the confidentiality of $\boldsymbol{o}^{(i)}$ or not is not clear. We have the theorem as follows,

Theorem 1. *For a volunteer N_i with observations $\boldsymbol{o}^{(i)} = \{\boldsymbol{o}_1^{(i)}, \boldsymbol{o}_2^{(i)}, \ldots, \boldsymbol{o}_{n_i}^{(i)}\}$, where $\boldsymbol{o}_j^{(i)}$ is a vector of $p+1$ dimensions. In order to defend against observation recovery attacks, the number of observations n_i should be more than $\frac{p}{2}+2$.*

Proof. An attacker aiming to recover $\boldsymbol{o}^{(i)}$ considers the problem as solving a set of equations, gathered during the procedure of 'clean' subset forming, related to the corresponding $n_i \times (p+1)$ unknown variables. According to the protocol, N_i publishes $\boldsymbol{s}^{(i)}$, $\boldsymbol{V}^{(i)}$ and several $d_M(\boldsymbol{o}_j^{(i)})$ (the number is $\min\{p+2, n_i\}$), revealing $p+1$, $\frac{(p+1)(p+2)}{2}$, and $\min\{p+2, n_i\}$ equations, respectively. It is essential that the number of variables should be larger than that of the corresponding equations, otherwise, $\boldsymbol{o}^{(i)}$ could be recovered. Note that N_i can shuffle the d_M before sharing, such that given $d_M(\boldsymbol{o}_\phi^{(i)})$, the attacker cannot decide the specific variable $\boldsymbol{o}_j^{(i)}$ associated with it, which means those Mahalanobis distances cannot be used to extend the equation set to recover $\boldsymbol{o}^{(i)}$. Thus, the following equality should be hold,

$$n_i \times (p+1) > (p+1) + \frac{(p+1)(p+2)}{2} \tag{5}$$

The condition for (5) is $n_i > \frac{p}{2} + 2$ and this concludes the proof. ■

2) Building Rough Global Model: suppose the local 'clean' set of N_i is $\boldsymbol{o}_c^{(i)} = \{\boldsymbol{o}_{c_1}^{(i)}, \ldots, \boldsymbol{c}_{c_i}^{(i)}\}$. According to (2), to build a rough regression model, N_i computes both $(\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)})$ and $(\boldsymbol{x}_c^{(i)})^T \boldsymbol{y}_c^{(i)}$, and submits the results to S. Unfortunately, sharing those values may cause privacy problem if the number of variables in $\boldsymbol{o}_c^{(i)}$ is no more than that of the equations obtained from $(\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)})$ and $(\boldsymbol{x}_c^{(i)})^T \boldsymbol{y}_c^{(i)}$.

To overcome this difficulty, we use a slicing technique for secure calculating the summation results of $\boldsymbol{u}_c = \sum_{i=1}^{m} (\boldsymbol{x}_c^{(i)})^T (\boldsymbol{x}_c^{(i)})$ and $\boldsymbol{v}_c = \sum_{i=1}^{m} (\boldsymbol{x}_c^{(i)})^T \boldsymbol{y}_c^{(i)}$, without disclosing individual $(\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)})$ and $(\boldsymbol{x}_c^{(i)})^T \boldsymbol{y}_c^{(i)}$.We explain the key idea of slicing in calculating as follows:

First, each N_i dynamically selects l_{in} other volunteers nearby to distribute its local result. It is assumed that any pair of volunteers nearby can achieve a unique pairwise key used for secure data transmission.

Second, N_i slices its data into $l_{in}+1$ random slices, i.e., $(\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)}) = \sum_{k=0}^{l_{in}} A_k^{(i)}$, where $A_k^{(i)}$ is a matrix of dimension $(p+1) \times (p+1)$.

Third, N_i keeps $A_0^{(i)}$ to itself while sending each other slice $A_k^{(i)}, k \neq 0$, to the corresponding volunteer N_l it has selected. Meanwhile, N_i can also receive l_{out} slices $A_i^{(k)}$ from l_{out} different volunteers. Then N_i recalculates its local matrix using its own slice $A_0^{(i)}$ and l_{out} slices received from others, i.e., $A_0^{(i)} + \sum_{k=1}^{l_{out}} A_i^{(k)}$, and sends it to S instead of $(\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)})$.

Finally, S adds up all the received values. It is easy to check that the result is exactly the sum of $\sum_{i=1}^{m} (\boldsymbol{x}_c^{(i)})^T(\boldsymbol{x}_c^{(i)})$.

After obtaining $\boldsymbol{u}_c$ and $\boldsymbol{v}_c$, it's convenient for S to calculate a rough model with coefficient vector $\widehat{\boldsymbol{\beta}}_{rgh}$ according to (2). We prove that it's unlikely for an attacker to recover $\boldsymbol{O}_c$ through $\boldsymbol{u}_c$ and $\boldsymbol{v}_c$ in subsection of Security Analysis.

3) Deriving Refined Global Model: After deriving rough model $\widehat{\boldsymbol{\beta}}_{rgh}$, S broadcasts $\widehat{\boldsymbol{\beta}}_{rgh}$ to all volunteers. Then **algorithm 2** is carried out between S and each N_i, in which the outlyingness of observations in $\boldsymbol{o}^{(i)}$ is tested in accordance with $\widehat{\boldsymbol{\beta}}_{rgh}$. Those observations fitting $\widehat{\boldsymbol{\beta}}_{rgh}$ well will be used to achieve a refined global estimate $\widehat{\boldsymbol{\beta}}_{ini}$.

Algorithm 2: Deriving refined global model

1: N_i calculates the Residual Sum of Squares (RSS) of $\boldsymbol{o}^{(i)}$ to $\widehat{\boldsymbol{\beta}}_{rgh}$ locally as $R_{ss}^{(i)} = (\boldsymbol{e}^{(i)})^T\boldsymbol{e}^{(i)}$, where $\boldsymbol{e}^{(i)} = [e_1^{(i)}, \ldots, e_{n_i}^{(i)}]^T$, and $e_j^{(i)} = y_j^{(i)} - \boldsymbol{x}_j^{(i)}\widehat{\boldsymbol{\beta}}_{rgh}$.
2: N_i sends $R_{ss}^{(i)}$ to S.
3: S calculates $R_{ss} = \sum_{i=1}^{m} R_{ss}^{(i)}$, and then broadcasts R_{ss} and $n = \sum_{i=1}^{m} n_i$ to $\mathbb{N}$.
4: N_i calculates standardized residual $z_j^{(i)}$ of $\boldsymbol{o}_j^{(i)}$, i.e., $z_j^{(i)} = |e_j^{(i)}|/\sqrt{Mse_j^{(i)}}$
5: N_i goes through $\boldsymbol{o}^{(i)}$, and labels observations with $z_j^{(i)}$ exceeding 1.69 according to [7] as outliers. Normal observations left are reformed as $\boldsymbol{o}_\eta^{(i)} = \{\boldsymbol{o}_{\eta_1}^{(i)}, \ldots, \boldsymbol{o}_{\eta_i}^{(i)}\}$. /*$\boldsymbol{o}_\eta^{(i)}$ is used to build the refined global model.*/
6: N_i computes $(\boldsymbol{x}_\eta^{(i)})^T(\boldsymbol{x}_\eta^{(i)})$ and $(\boldsymbol{x}_\eta^{(i)})^T\boldsymbol{y}_\eta^{(i)}$ locally, and submits them to S by using slicing.
7: S calculates $\boldsymbol{u}_\eta = \sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T(\boldsymbol{x}_\eta^{(i)})$ and $\boldsymbol{v}_\eta = \sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T\boldsymbol{y}_\eta^{(i)}$, and re-estimates $\widehat{\boldsymbol{\beta}}_{ini} = (\boldsymbol{u}_\eta)^{-1}\boldsymbol{v}_\eta$.

In the above algorithm, $Mse_j^{(i)} = \frac{(R_{ss} - (e_j^{(i)})^2)}{(n-p-2)}$ is the mean squared error excluding $e_j^{(i)}$, and η_i equals to n_i minus the number of outliers in $\boldsymbol{o}^{(i)}$.

Notice that S obtains $\boldsymbol{u}_\eta$ and $\boldsymbol{v}_\eta$, related to $\boldsymbol{O}_\eta = \bigcup_{i=1}^{m}\boldsymbol{o}_\eta^{(i)}$. In order to guarantee that $\boldsymbol{O}_\eta$ will not be recovered by S or other attackers. We have the theorem as follows,

Theorem 2. *In order to defend against $\boldsymbol{O}_\eta$ from been recovered by S or other attackers, the number of volunteers m should be at least 6.*

Proof. An attacker aiming to recover $\boldsymbol{O}_\eta$ considers the problem as solving a set of equations related to the corresponding observations. The attacker can obtain the information about the number of observations in $\boldsymbol{O}_\eta$, denoted by x (during the coming model updating, in order to calculate Mahalanobis distances of new observations, the number of new observations α should be known to S. Meanwhile, in order to calculate $Mse_j^{(i)}$, S obtains the total number of observations in the model updating, i.e., $x + \alpha$. Thus it's easy to compute x). As each observation contains $p+1$ unknown variables, the number of unknown variables is $x(p+1)$. By gathering $\boldsymbol{u}_\eta$ and $\boldsymbol{v}_\eta$, the attacker can establish $(p+1)+\frac{(p+1)(p+2)}{2}$ equations about $\boldsymbol{O}_\eta$. Consider the worst case that all observations in $\boldsymbol{O}_c$ belong to $\boldsymbol{O}_\eta$, it implies that the attacker can calculate $\boldsymbol{u}_\eta - \boldsymbol{u}_c$ and $\boldsymbol{v}_\eta - \boldsymbol{v}_c$, which only relate to $x - (p+2)$ observations. Thus, the following equality should be hold,

$$(p+1) + \frac{(p+1)(p+2)}{2} < (p+1)[x - (p+2)] \quad (6)$$

i.e., $\frac{3p}{2} + 4 < x$. Consider that there should be at most half outliers in all observations, which ensures $x > \frac{n}{2}$ can be satisfied, then we require that $\frac{3p}{2} + 4 < \frac{n}{2}$ holds. Since each N_i has at least $\frac{p}{2} + 2$ observations, which implies $m(\frac{p}{2} + 2) < n$, we require $\frac{3p}{2} + 4 < \frac{m}{2}(\frac{p}{2} + 2)$ holds. Thus $\frac{6p+16}{p+4} < \frac{6(p+4)}{p+4} \leq m$, i.e., $6 \leq m$ can satisfy the condition for (7) and this concludes the proof. ■

C. Adaptive Model Updating

Under the current estimate $\widehat{\boldsymbol{\beta}}_{ini}$, assume that there exists a group of volunteers $\widetilde{\mathbb{N}} = \{\widetilde{N}_1, \ldots, \widetilde{N}_q\}$, and each $\widetilde{N}_k$ $(1 \leqslant k \leqslant q)$ holds a set of new observations, i.e., $\widetilde{\boldsymbol{o}}^{(k)} = \{\widetilde{\boldsymbol{o}}_1^{(k)}, \ldots, \widetilde{\boldsymbol{o}}_{q_k}^{(k)}\}$, where q_k denotes the number of observations of $\widetilde{N}_k$. In order to update $\widehat{\boldsymbol{\beta}}_{ini}$ using new observations, the server S first calculates a new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$ according to the current estimate $\widehat{\boldsymbol{\beta}}_{ini}$ and the new 'clean' subset, and then refines $\widetilde{\boldsymbol{\beta}}_{rgh}$ to achieve the new estimate.

1) Calculating New Rough Model: A new size $p+2$ 'clean' subset is formed and included into $\widehat{\boldsymbol{\beta}}_{ini}$.

First, S and $\widetilde{N}_i$ cooperate with each other to select the new 'clean' subset (with a size of $p+2$) from $\widetilde{\boldsymbol{O}} = \{\widetilde{\boldsymbol{o}}^{(1)}, , \ldots, \widetilde{\boldsymbol{o}}^{(q)}\}$, following the protocol which has been introduced to find the basic 'clean' subset. We denote the new 'clean' observation subset of $\widetilde{N}_k$ as $\widetilde{\boldsymbol{o}}_a^{(k)} = \{\widetilde{\boldsymbol{o}}_{a_1}^{(k)}, \ldots, \widetilde{\boldsymbol{o}}_{a_k}^{(k)}\}$.

Second, each $\widetilde{N}_k$ uses slicing for secure aggregating $\sum_{k=1}^{q}(\widetilde{\boldsymbol{x}}_a^{(k)})^T(\widetilde{\boldsymbol{x}}_a^{(k)})$ and $\sum_{k=1}^{q}(\widetilde{\boldsymbol{x}}_a^{(k)})^T\widetilde{\boldsymbol{y}}_a^{(k)}$ on the server side, without disclosing individual $(\widetilde{\boldsymbol{x}}_a^{(k)})^T(\widetilde{\boldsymbol{x}}_a^{(k)})$ and $(\widetilde{\boldsymbol{x}}_a^{(k)})^T\widetilde{\boldsymbol{y}}_a^{(k)}$.

Third, since that $\sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T(\boldsymbol{x}_\eta^{(i)})$ and $\sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T\boldsymbol{y}_\eta^{(i)}$ have been computed to achieve $\widehat{\boldsymbol{\beta}}_{ini}$, S adds the corresponding aggregation results together, i.e.,

$$\widetilde{\boldsymbol{u}}_{rgh} = \sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T(\boldsymbol{x}_\eta^{(i)}) + \sum_{k=1}^{q}(\widetilde{\boldsymbol{x}}_a^{(k)})^T(\boldsymbol{x}_a^{(k)})$$

$$\widetilde{\boldsymbol{v}}_{rgh} = \sum_{i=1}^{m}(\boldsymbol{x}_\eta^{(i)})^T\boldsymbol{y}_\eta^{(i)} + \sum_{k=1}^{q}(\widetilde{\boldsymbol{x}}_a^{(k)})^T\widetilde{\boldsymbol{y}}_a^{(k)}$$

then calculates $\widetilde{\boldsymbol{\beta}}_{rgh} = (\widetilde{\boldsymbol{u}}_{rgh})^{-1}\widetilde{\boldsymbol{v}}_{rgh}$.

2) Deriving New Refined Model: After updating the new rough model $\widetilde{\boldsymbol{\beta}}_{rgh}$, S broadcasts it to $\mathbb{U} = \widetilde{\mathbb{N}} \bigcup \mathbb{N}$, where $\mathbb{N}$ refers to the set of volunteers whose observations have been included in $\widehat{\boldsymbol{\beta}}_{ini}$. Then the following procedures are performed to establish a new model $\widetilde{\boldsymbol{\beta}}_{new}$.

First, S communicates with each volunteer in $\mathbb{U}$, to test the outlyingness of observations in $\boldsymbol{O}_\eta \bigcup \widetilde{\boldsymbol{O}}$, in accordance to $\widetilde{\boldsymbol{\beta}}_{rgh}$, which is alike the procedure obtaining $\boldsymbol{o}_\eta^{(i)}$ (described in subsection III-B3).

Those observations unfitting $\widetilde{\boldsymbol{\beta}}_{rgh}$ will be removed from the new model, which implies the size of $\boldsymbol{o}_\eta^{(i)}$. While observations in $\widetilde{\boldsymbol{o}}^{(k)}$ passing the test will be added into the new model.

To this end, assume that a set of observations $\boldsymbol{O}_\tau = \bigcup_{i=1}^{m}\boldsymbol{o}_\tau^{(i)}$, where $\boldsymbol{o}_\tau^{(i)} \subseteq \boldsymbol{o}_\eta^{(i)}$ should be removed (the left part is denoted as $\boldsymbol{O}_\varphi = \bigcup_{i=1}^{m}\boldsymbol{o}_\varphi^{(i)}$, where $\boldsymbol{o}_\varphi^{(i)} = \boldsymbol{o}_\eta^{(i)} - \boldsymbol{o}_\tau^{(i)}$), and $\widetilde{\boldsymbol{O}}_\eta = \bigcup_{k=1}^{q}\widetilde{\boldsymbol{o}}_\eta^{(k)}$, where $\widetilde{\boldsymbol{o}}_\eta^{(k)} \subseteq \widetilde{\boldsymbol{o}}^{(k)}$, should be included, then

$$\boldsymbol{X}_{new} = \begin{bmatrix}\boldsymbol{X}_\varphi \\ \widetilde{\boldsymbol{X}}_\eta\end{bmatrix} \boldsymbol{Y}_{new} = \begin{bmatrix}\boldsymbol{Y}_\varphi \\ \widetilde{\boldsymbol{Y}}_\eta\end{bmatrix} \boldsymbol{X}_\eta = \begin{bmatrix}\boldsymbol{X}_\varphi \\ \boldsymbol{X}_\tau\end{bmatrix} \boldsymbol{Y}_\eta = \begin{bmatrix}\boldsymbol{Y}_\varphi \\ \boldsymbol{Y}_\tau\end{bmatrix}$$

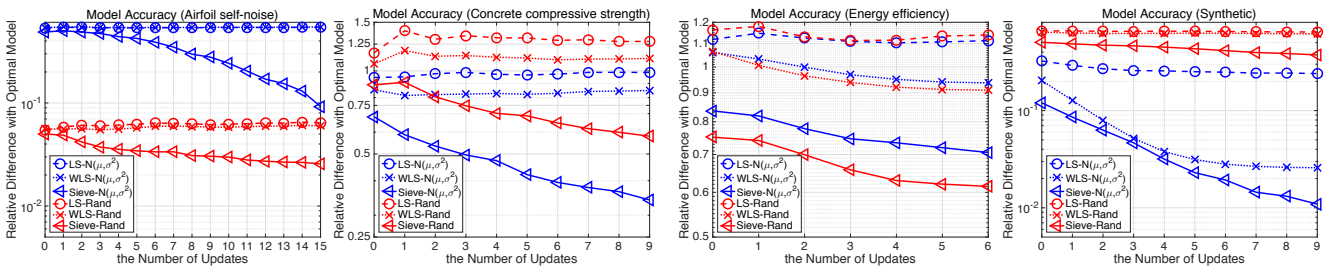

Fig. 2. Difference from the optimal model vs. the number of updates.

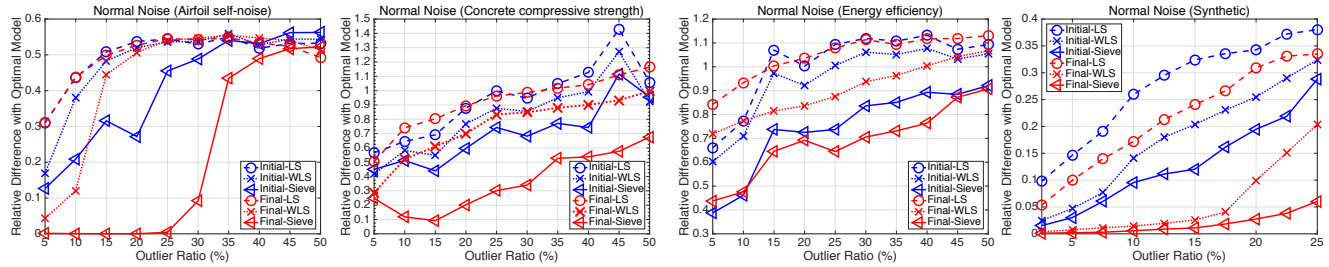

Fig. 3. Difference from the optimal model vs. outlier ratio with normal noise under $N(\mu, \sigma^2)$.

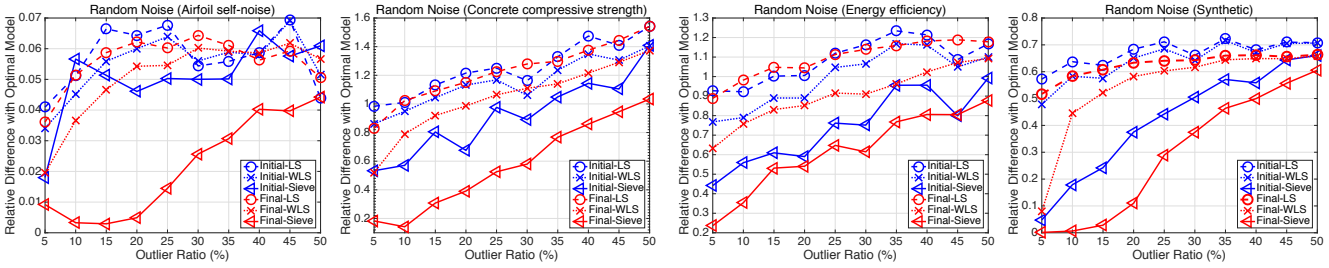

Fig. 4. Difference from the optimal model vs. outlier ratio with random noise.

Consider that

$$\widetilde{\boldsymbol{u}}_{new} = \boldsymbol{u}_\eta + \sum_{k=1}^{q} (\widetilde{\boldsymbol{x}}_\eta^{(k)})^T (\widetilde{\boldsymbol{x}}_\eta^{(k)}) - \sum_{i=1}^{m} (\boldsymbol{x}_\tau^i)^T (\boldsymbol{x}_\tau^{(i)})$$

$$\widetilde{\boldsymbol{v}}_{new} = \boldsymbol{v}_\eta + \sum_{k=1}^{q} (\widetilde{\boldsymbol{x}}_\eta^{(k)})^T (\widetilde{\boldsymbol{y}}_\eta^{(k)}) - \sum_{i=1}^{m} (\boldsymbol{x}_\tau^{(i)})^T (\boldsymbol{y}_\tau^{(i)})$$

then $\widetilde{\boldsymbol{\beta}}_{new} = (\widetilde{\boldsymbol{u}}_{new})^{-1} \widetilde{\boldsymbol{v}}_{new}$. Notice that aggregation results should be shared using slicing for privacy consideration.

IV. Performance Evaluation

A. Methodology

We verify the performance of Sieve via the following three real datasets and one synthetic dataset,

1) *Airfoil self-noise* (1503 observations) [8]: *three* attributes, i.e., *frequency*, *angle of attack* and *free-stream velocity*, are used.
2) *Energy efficiency* (768 observations) [9]: *six* attributes, i.e., *surface relative compactness*, *orientation*, *roof area*, *overall height*, *glazing area*, and *glazing area distribution*, are selected.
3) *Concrete compressive strength* (1030 observations) [10]: *four* attributes, i.e., *cement*, *blast furnace slag*, *fly ash* and *age*, are adopted.
4) *Synthetic*: we generate 1400 observations by using $y = 5 + 5\sum_{i=1}^{9} x_i + \epsilon$, where $\epsilon \sim N(0,1)$ and $x_i \sim N(0,1)$.

We consider following two kinds of noises:

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 Sieve with the state-of-art LS estimator, and WLS estimator proposed in PURE [4].

By utilizing LS estimator, we can obtain the models to the original datasets (without noises), which are high quality, as the ground truth. 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 simulation configuration, we run the simulation 20 times and get the average.

B. Accuracy with Model Updating

For simulating periodical regression model updating, 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 S. Green [11]. each group will only be used one time for either initial model estimating or updating. We compare Sieve with LS and WLS, under both noise settings with the outlier ratio ε equals to 30% in each observation group. Fig. 2 plots the model accuracy as a function of the number of updates, where *zero* indicates the initial model. Note that the plots are on a linear-log scale. It can be seen that model accuracy estimated by Sieve is continuously improved with new group of observations being included in model updating, and Sieve outwits its comparisons in all settings of real datasets.

C. Robustness under Different Outlier Ratios

We study the robustness of Sieve against outliers under different outlier ratios ε and types. We examine both initial estimates, and final models obtained after all rounds of updating. For each kind of noise, the noise ratio ε varies from 5% to 50% at an interval of 5%. 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. 3 and Fig. 4 plot the model accuracy as a function of outlier ratio under random and normal distribution noise settings, respectively. Fig. 3 and Fig. 4 demonstrate that Sieve outwits LS and WLS in all settings. It can be seen that LS is very sensitive to outliers, whereas, Sieve maintains good accuracy even when ε increases to 40%. WLS may achieve a high accuracy of final models under low outlier ratio, however, the performance degrades dramatically as ε increasing to 10%. Oppositely, Sieve is much more tolerant to outliers, and model updating can significantly improve the accuracy of estimates.

V. Related Work

An efficient way of protecting private data is perturbation, the core idea of which is to add the random noise with known distribution to raw data, and to run a reconstruction algorithm to estimate the distribution of original data. For example, R. Agrawal et al. [12] proposed a perturbation-based method enabling privacy-preserving multidimensional aggregation on data across multi-agents. The shortage of perturbation-based schemes is that additional noises introduced make regression estimates inaccurate. Some distanced-based outlier detection schemes have been investigated [1], [2]. However, the definition of outliers in regression is much more complex.

Du et al. [13] addressed the problems of Secure two-party Multivariate Linear Regression and Classification. I. Giacomelli et al. introduced a homomorphic encryption-based regression scheme [14]. M-PERM is a mutual privacy-preserving regression approach, where a series of data transformations and aggregations are operated at the participatory nodes to preserve data privacy [3]. However, all these methods utilized LS estimator. The correctness of them relies on the assumption that original data are collected correctly. Any outlier may breakdown the estimation, since LS estimator is very sensitive. PURE, an outlier tolerant privacy-preserving regression scheme is most relevant to our work [4]. However, PURE failed to 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.

VI. Conclusion

We proposed Sieve, a scheme for lightweight robust regression on low-quality, private sensory data in MCS. In Sieve, volunteers publish aggregated value of private observations which are proven secure, preventing private observations from being revealed by other. Moreover, Sieve can identify suspected outliers and withdraw outdated observations during model updating accurately, maintaining high model accuracy under low quality and non-stationary data. Extensive simulation results demonstrate the efficacy of Sieve.

Acknowledgment

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 (2019), DHU Distinguished Young Professor Program and 2017 CCF-IFAA Research Fund.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] P. Mahalanobis, "On the generalised distance in statistics," in *Proc. NISI*, 1936, pp. 49-55.

[6] P. J. Rousseeuw and B. C. van Zomeren, "Unmasking multivariate outliers and leverage points," in *American Statistical Association*, 1990, pp. 633-639.

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

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

[9] Energy efficiency dataset.[Online]. Availiable: https://archive.ics.uci.edu/ml/datasets/Energy+efficiency.

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

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

[12] R. Agrawal, R. Srikant, and D. Thomas, "Privacy preserving OLAP," in *Proc.ACM SIGMOD*, 2005, pp. 251-262.

[13] W. Du, Y. Han and S. Chen, "Privacy-preserving multivariate statistical analysis: linear regression and classification," in *Proc. EECS*, 2004, pp. 222-233.

[14] I. Giacomelli, S. Jha, and C. D. Page, "Privacy preserving linear regression on distributed data," in *Cryptology ePrint Archive*, 2017, pp. 1-19.