RR-LADP: A Privacy-Enhanced Federated Learning Scheme for Internet of Everything

Article *in* IEEE Consumer Electronics Magazine · February 2021

DOI: 10.1109/MCE.2021.3059958

7 authors, including:

Yang Liu
Harbin Institute of Technology (Shenzhen)
57 PUBLICATIONS **330** CITATIONS
SEE PROFILE

Mohsen Guizani
Mohamed bin Zayed University of Artificial Intelligence
1,178 PUBLICATIONS **53,154** CITATIONS
SEE PROFILE

RR-LADP: A Privacy-Enhanced Federated Learning Scheme for Internet of Everything

Zerui Li
Harbin Institute of Technology, Shenzhen

Yuchen Tian
Harbin Institute of Technology, Shenzhen

Weizhe Zhang
Harbin Institute of Technology, Shenzhen
Peng Cheng Laboratory

Qing Liao
Harbin Institute of Technology, Shenzhen

Yang Liu
Harbin Institute of Technology, Shenzhen
Peng Cheng Laboratory

Xiaojiang Du
Temple University, Philadelphia

Mohsen Guizani
Qatar University, Doha

***Abstract*—While the widespread use of ubiquitously connected devices in IoE offers enormous benefits, it also raises serious privacy concerns. Federated learning, as one of the promising solutions to alleviate such problems, is considered as capable of performing data training without exposing raw data that kept by multiple devices. However, either malicious attackers or untrusted servers, can deduce users' privacy from the local updates of each device. Previous studies mainly focus on privacy-preserving approaches inside the servers, which requires the framework to be built on trusted servers. In this paper, we propose a privacy-enhanced federated learning scheme for IoE. Two mechanisms are adopted in our approach, namely the randomized response (RR) mechanism and the local adaptive differential privacy (LADP) mechanism. RR is adopted to prevent the server from knowing whose updates are collected in each round. LADP enables devices to add noise adaptively to its local updates before submitting them to the server. Experiments demonstrate the feasibility and effectiveness of our approach.**

I. Introduction

THE Internet of Everything (IoE) redefines the connection between people, things and data and changes the way to interact devices. In this era, any object can be transformed into network data through corresponding sensors. At the same time, advances in communication technology and enhancement of edge computing capabilities facilitate the application of machine learning in the Internet of Everything, which means that network data can be effectively mined to support intelligent services. For example, by collecting static and dynamic information of users, smart furniture and intelligence software can improve the efficiency of work and life. In general, more data means better service. In this environment, most IoT devices continuously collect and upload private data during operation, which is potential to compromise users' privacy [**?**].

Recently, federated learning, which localizes the training process, is seen as a potential machine learning mechanism to solve the problem of using private data. It distributes the tasks of model training to multiple participants and aggregates local updates to iteratively generate a global model. The advantage is that the information delivered to server is model weights difference or training gradient [**?**], instead of the raw data. This distributed learning setup decouples the training tasks from the need for the server to centralize data. It can use the private data of different users to learn a high-quality sharing model, while leaving raw data on the local IoT devices. Therefore, the risk of privacy leakage caused by data transmission, cloud storage, and centralized training can be effectively decreased. At the same time, the availability of training data is guaranteed as there is no need to encrypt the data before using it.

In fact, similar to most privacy solutions, federated learning is also based on an important assumption that the process is scheduled by a trusted server. Untrusted servers and malicious attackers may perform model inversion attacks by obtaining the communication parameters of federated learning. In the IoE solutions, data collected from sensors will be transmitted through multiple layers to generate services. The multi-layer structure composed of hardware and software makes federated learning more vulnerable to potential attacks.

In this paper, we propose a privacy-enhanced federated learning scheme for IoE. Two mechanisms are adopted in our approach, namely the randomized response (RR) mechanism and the local adaptive differential privacy (LADP) mechanism. The main contributions are as follows.

- We propose the RR mechanism, which is completed by each device to enhance the privacy of devices selection. In each training round of RR federated learning, the server cannot determine whether a particular device is participated the training. The mechanism, therefore, can prevent untrusted servers and malicious attackers from knowing which devices' updates are included in the communication content.
- We adopt the LADP mechanism in the stage of local training. Gaussian noise is added to the local updates of each device adaptively before the the updates are uploaded to the server. Hence, the mechanism can even prevent untrusted server and malicious attackers from deducing relevant information of the training data with local updates.

II. Background and motivation

A. Federated learning

The general flow of federated learning using FederatedAveraging (FedAvg) algorithm to aggregate updates is as follows [**?**].

Suppose there are a total of K clients and each client has a private dataset. In each training round t, the server randomly selects $K^{'}(K^{'} \leq K)$ clients and sends them the global model with the weight ω_{t-1}. Each client k selected trains the model on its private data, and uploads the weights difference $\Delta\omega_t^k$. Finally, the server averages these local updates and generates a new global model, and the process repeats as:

$$\omega_t = \omega_{t-1} + \frac{1}{K^{'}} \sum \Delta\omega_t^k \tag{1}$$

B. Differential privacy

Differential privacy provides a strong privacy guarantee for aggregate data [**?**]. Its definition is as follows.

Define two datasets to be adjacent if they differ only in a single record. A given mechanism $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{R}$ has domain $\mathcal{D}$ and range $\mathcal{R}$. We define the mechanism $\mathcal{M}$ satisfies ε-differential privacy, if for any two adjacent inputs $d, d^{'} \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \in \mathcal{R}$ the following inequality holds:

$$Pr[\mathcal{M}(d) \in \mathcal{S}] \leq e^{\varepsilon} Pr[\mathcal{M}(d^{'}) \in \mathcal{S}] + \delta \tag{2}$$

The privacy budget ε limits the bounds of privacy loss and the slack variable δ allows the definition break with a given probability.

The general way to realize this mechanism is to add Gaussian noise to approximate a real value function $f : \mathcal{D} \rightarrow \mathcal{R}$ with differential privacy. The noise is calibrated to sensitivity S_f, which is the maximum value of absolute distance $|f(d) - f(d^{'})|$. $f(d)$ and $f(d^{'})$ are function value corresponding to the adjacent input d and $d^{'}$. We define a Gaussian noise addition mechanism as $\mathcal{M}(d) = f(d) + \mathcal{N}(0, {S_f}^2\sigma^2)$, where $\mathcal{N}(0, {S_f}^2\sigma^2)$ is the Gaussian distributed noise with mean 0 and standard deviation $S_f\sigma$.

C. Motivation

The privacy-preserving federated learning can be realized by leveraging differential privacy [**?**], which effectively reduce the possibility to infer extra information through data transferred in each training round. Ideally, it requires a trusted server to complete the noise addition operation. In the real world training process, we consider the following situations:

- **The server is curious:** The server can normally complete the privacy processing steps such as noise addition after FedAvg. At the same time, it wants to infer the private data of clients.

- **The server is incompetent:** The server may fail to add noise into the averaged updates for some reason before releasing the global model. It puts all participating clients under great risks of privacy leakage.

Due to the deficiency of centralized privacy-preserving approach, we adjust the client selection mechanism and shifting the noise addition to client side in order to reduce the dependence on the server. In the context of FedAvg, we are more interested in the weights difference contributed by each client. In this way, the noise contained in the aggregation is the sum of the noise added by each client. It will satisfies differential privacy if each client process satisfies [**?**].

III. State-of-the-Art

Instead of collecting data from clients and training the model on the server in a centralized way, federated learning allows multiple clients to learn a model collaboratively while keeping data locally. It provides a new solution for preserving privacy in machine learning. Google first proposed federated learning [**?**], a privacy-preserving collaborative modeling mechanism. They applied federated learning to the input prediction and query suggestions of Gboard [**?**], [**?**]. Konecny et al. [**?**] used structured updates and model compression to reduce uplink and downlink communication costs. Bonawitz et al. proposed a protocol user to improve the robustness of federated learning [**?**]. However, they did not consider the privacy risks of the federated learning mechanism.

The research of Fredrikson et al. [**?**] show that after training, sample data involved in the model training can be reconstructed via model parameters, even if data is remained locally. To minimize such disclosure, Geyer et al. [**?**] incorporated differential privacy in the aggregation update on the server side. Differential privacy can indeed reduce the correlation between final model and aggregated updates. However, it is a post-processing method with some limitations. Dealing with the results of FedAvg directly and ignoring the training process may reduce the usability of the model and increase the difficulty of observing the true expression of raw data for noises. Moreover, such approach ignores the protection of the updates transmitted during the communication, making the model vulnerable to inversion attacks. Agarwal et al. [**?**] proposed to add noise distributedly to approximate global privacy. Wei et al. [**?**] applied this method in federated learning. The global model will satisfy differential privacy when each part satisfies. However, the noise addition is performed by the server. Such privacy preservation is invalid for untrusted servers. Some researchers [**?**], [**?**] incorporated homomorphic encryption into federated learning. For such encryption based method, local updates are transmitted and calculated in the form of ciphertext. Therefore, it can well preserve the privacy without losing model accuracy. However, the calculation types supported by homomorphic encryption are limited. For ciphertext, the encryption/decryption and the calculation of it are with great computation overhead [**?**]. The transmission of it is also time consuming. These limitations make the approach impractical to be used in IoE.

IV. Proposed Method

A. The system design

Considering the diversity of devices, we uniformly term the actual devices in IoE as clients for the convenience of analysis. Fig. 1 illustrates the main components and process of RR-LADP. For a collaborative learning task, the server distributes the initial model and samples clients. Then each client randomly responds to the training request. After that, the actual participants train the model locally with their private data, and appropriately add noise to weights difference for privacy preserving. Finally, an edge computing node (i.e., edge server with secure multi-party summation protocol) aggregates all local updates and returns the result to the server to update global model. Through several rounds, the global model that incorporates contributions from multiple clients can perform well.

B. RR federated learning

Due to the huge number of devices in IoE environment, it is impracticable to involve all devices that meet the training requirements in each training round. Training tasks involving a large number of devices is more likely to be interrupted, which poses a great challenge to the distributed decision

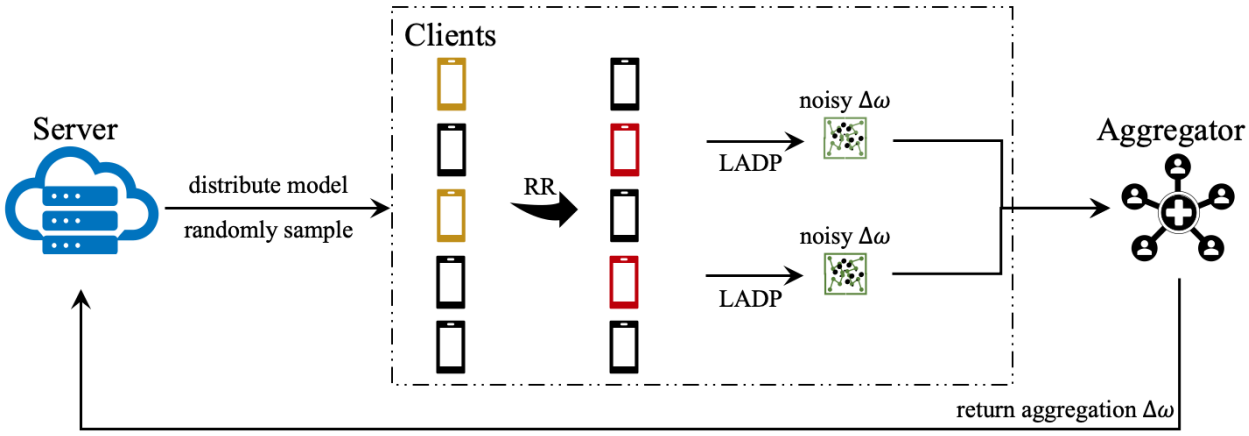

Fig. 1: The RR-LADP Framework.

making capabilities of the server [**?**], [**?**]. In fact, only a fraction of them will suffice to generate a desirable model, and in this way, the communication pressure can be effectively reduced. However, in traditional federated learning flow, the server masters the whole process of client selection. We propose a disturbance mechanism termed as RR, which could cause some deviation between actual participants and server sampling results. Therefore, it is hard for the curious server and attackers to correspond the final model to a particular client.

Before each round of communication, the server checks and establishes communication with IoT devices which meet the training requirements. Suppose a total of K eligible clients participate in global model construction. Server randomly selects $K^{'}(K^{'} \leq K)$ clients for training. We define a state parameter λ_t^k with value of 0 or 1 represents whether client k participates in round t, and a response probability p. Based on server sampling results, clients initialize their state parameter. All clients keep their state parameters unchanged with probability p and flip them with probability $(1-p)$. Then, we can calculate participation probability of each client k as follows:

$$Pr[\lambda_t^k = 1] = \frac{K^{'}}{K} * p + (1 - \frac{K^{'}}{K}) * (1 - p) \quad (3)$$

According to equation 3, server can estimate the number of participants with $\hat{K}_t = Pr[\lambda_t^k = 1] * K$. By constructing a maximum likelihood function, we can verify that $\hat{K}_t$ is the unbiased estimate of K_t, which is the number of actual participants in round t. In this way, server can get a value that is similar to K_t for FedAvg [**?**] in each round.

Additionally, we set $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}$ to satisfy ε-differential privacy so that RR federated learning can meet rigorous rather than intuitive privacy guarantees [**?**]. The response probability p and privacy budget ε are positively correlated. The higher the privacy budget, the more likely selected clients response, which means that actual participation is similar to the server sampling results. It may cause the risk of privacy leakage. Conversely, a lower privacy budget leads to a higher flip probability and lower risk.

C. Local adaptive differential privacy

Centralized privacy enhancing process has certain risks, because the local updates of each participant can be obtained before the aggregation. According to the composition theorems in [**?**], the global process can satisfy (ε, δ)-differential privacy, if each local process satisfies (ε_k, δ)-differential privacy and $\sum_{k=1}^{K} \varepsilon_k \leq \varepsilon$. Therefore, we consider incorporating differential privacy into the client. The following details our solution based on FedAvg.

Algorithm 1 gives the basic process of LADP. For each responding client, we train the weights matrix to get the difference $\Delta\omega_t^k = \omega_t^k - \omega_{t-1}$ from the global model generated by the last round. In each epoch, we optimize loss function by batch gradient descent (BGD) and record the 2-norm of the difference matrix $\|\omega - \omega_{t-1}\|_2$. When epochs

Algorithm 1 Local update of LADP

update(k, ω_{t-1})
  $initialize\ \lambda_t^k$
  if $\lambda_t^k = 1$ **then**
    $\omega \leftarrow \omega_{t-1}$
    $\mathcal{B} \leftarrow split\ \mathcal{S}et_k\ into\ batches$
    for $each\ local\ epoch\ i = 1, 2, 3, ...$ **do**
      for $batch\ b \in \mathcal{B}$ **do**
        $\omega \leftarrow \omega - \eta \nabla \mathcal{L}(\omega, b)$
      $C_i \leftarrow \|\omega - \omega_{t-1}\|_2$
    $C = \overline{C_i}(i = 1, 2, 3, ... \frac{|\mathcal{B}|}{|b|})$
    $\Delta\omega \leftarrow (\omega - \omega_{t-1} + \frac{1}{|\mathcal{B}|}\mathcal{N}(0, \sigma^2 S^2))$
    $\Delta\omega \leftarrow \Delta\omega \cdot min(1, \frac{C}{\|\Delta\omega\|_2})$
  else
    $\Delta\omega \leftarrow 0$
  $return\ \Delta\omega$

reach the upper bound, we stop training and clip difference matrix with C, which is the mean of the norms. If $\|\omega - \omega_{t-1}\|_2 < C$, keep elements of difference matrix unchanged. Otherwise scale down the elements with $\frac{C}{\|\Delta\omega\|_2}$. It can effectively reduce the expression of private data and improve the generalization ability of global model. Before sending updates to the aggregator, we add noise to it to enhance privacy.

We adopt the Gaussian mechanism distort local updates of each client. Noise variance $\sigma^2 S^2$ determines the retention of contributions from clients. Excessive noise means updates is highly distorted, but less noise cannot meet the privacy preserving. In each training, σ is fixed, and the values of S will be adjusted adaptively. On the one hand, we set $S = C$ to adjust the noise addition according to the updates itself. If a single updates is outstanding, the noise will increase. On the other hand, we expect the clients with different amounts of data could contribute similarly to the global model. Thus, we scale down the noise with $\frac{1}{|\mathcal{B}|}$.

D. Track privacy loss globally

Privacy loss reflects the risk of data privacy leakage. We adopt moments accountants [**?**] to track and limit privacy loss. In model training with multiple rounds, we consider the knowledge inheritance. Suppose ξ is the observation result of adjacent datasets d and $d^{'}$ under $\mathcal{M}$, we define privacy loss with $\mathcal{L}^{(\xi)}_{\mathcal{M}(pre,d)||\mathcal{M}(pre,d')} = \ln \frac{Pr[\mathcal{M}(pre,d)=\xi]}{Pr[\mathcal{M}(pre,d')=\xi]}$. The pre is including all previous outputs. Privacy loss increases if the probability that the observation comes from the original set is higher. We define $\alpha^{(\tau)}_{\mathcal{M}(pre,d)||\mathcal{M}(pre,d')}$ as the cumulant generating function of $\mathcal{L}^{(\xi)}_{\mathcal{M}(pre,d)||\mathcal{M}(pre,d')}$ at value τ. Considering all the adjacent datasets and all possible previous outputs, we track the privacy loss of client k as follows:

$$\alpha^{(\tau)}_{\mathcal{M}_k} \triangleq \max_{pre,d,d'} \alpha^{(\tau)}_{\mathcal{M}_k(pre,d)||\mathcal{M}_k(pre,d')} \tag{4}$$

Then we can track the global privacy loss with $\alpha^{(\tau)}_{\mathcal{M}} = \sum \alpha^{(\tau)}_{\mathcal{M}_k}$. For any fixed privacy budget ε, we can calculate the current value of slack variable with $\delta = \min_{\tau} e^{\alpha^{(\tau)}_{\mathcal{M}} - \tau\varepsilon}$. When δ reaches bound, the accumulated privacy loss after current round is out of tolerance. Thus, we stop training and return the result. The setting of bound usually depends on the sample space. Considering the disturbance from both RR and LADP, we set the bound to $\frac{1}{|K\mathcal{B}|}$.

V. Experiment and analysis

We simulate the training process of federated learning and apply our proposed mechanism to it. These experimental results verified the feasibility and effectiveness of our proposed mechanism. Each client trains a fully connected neural network with the same structure, which contains two hidden layers with 600 and 400 neurons. The simple neural network structure allows us to better evaluate the impact of the mechanism. Cross entropy is chosen as the loss function. The learning rate is 0.1 and the optimization method perform within clients is BGD. In order to simulate non-IID distributed data, we divide MNIST into different subsets, and each of them contains only two or three digit samples. A model trained on the dataset of a single client cannot accurately recognize all digits. Before training, each client divide its dataset into multiple batches $\mathcal{B} = \{b_1, b_2, b_3, ...\}$. For comparison, the value of $K^{'}$ follows the CDP setting [**?**], that is $K^{'} = 30, 100, 300$ when $K = 100, 1000, 10000$. Similarly, the batch size is set to 10 and the number of epochs trained by each client is 4.

Separate performance: In the early stages of the experiment, we evaluate the effects of RR and LADP separately. To verify the feasibility of RR,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2021.3059958, IEEE Consumer Electronics Magazine

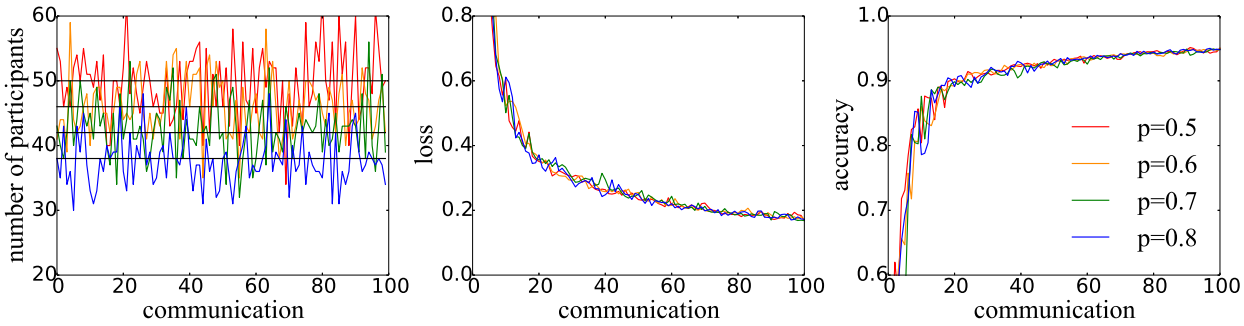

Fig. 2: Training process of RR federated learning. (We do not apply privacy bound here, but set the max round to 100. The legend in plot 3 also applies to the other two plots.)

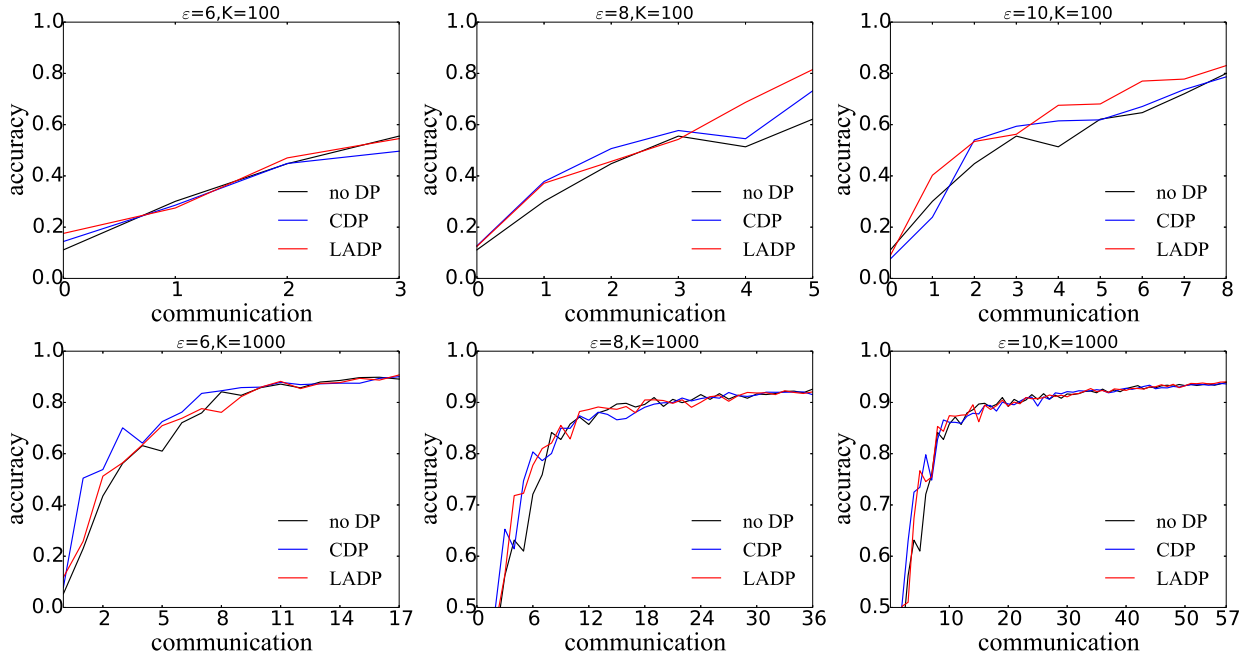

Fig. 3: Results on the Accuracy for LADP and CDP. ($\sigma = 1$)

we conducted experiments on RR federated learning under different disturbances, which depends on ε, when $K = 100$ and $K^{'} = 30$. Although the actual number of participants fluctuates around estimated result, it does not have consequences on the normal convergence and accuracy of the global model(see Fig. 2). At the same time, we design the comparative experiment of LADP and CDP to observe the actual performance of LADP. Fig. 3 illustrates that the accuracy of LADP rises smoothly and performs well with different privacy budgets, especially when the participants are few and the privacy budget is low. LADP directly adds noise to difference matrices $\Delta\omega$. Suppose the noise is ζ. We can get $\Delta\omega + \zeta = \eta(\nabla\mathcal{L} + \frac{\zeta}{\eta})$. It is equivalent to disturbing gradient so that the influence on the final model is traceable and controllable. However, CDP deals with the average, and treats the training process of each client as a black box. Then, its performance usually fluctuates.

Comparison with CDP: To compare RR-LADP and CDP, we track privacy budget in each mechanism by calculating δ. In fact, accumulated knowledge inheritance pre makes the privacy loss of each round increase rapidly. Saving privacy budget of the same order of magnitude can not support

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/MCE.2021.3059958, IEEE Consumer Electronics Magazine

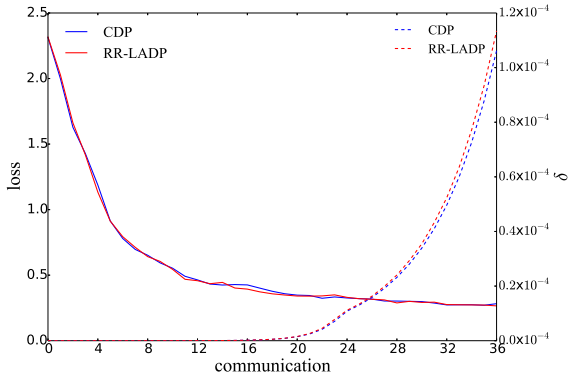

Fig. 4: Training process of RR-LADP and CDP when $K = 1000$ and $K^{'} = 100$. The solid lines represent loss and the dotted lines represent δ. $(\varepsilon = 8, \sigma = 1)$

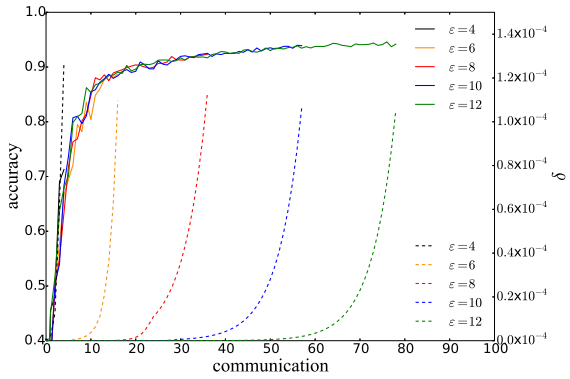

Fig. 5: Results on the accuracy and δ for different privacy budgets. The solid lines represent accuracy and the dotted lines represent δ. $(\sigma = 1, K = 1000, K^{'} = 100)$

more training rounds. Therefore, careful allocation means efficient data usage. Fig. 4 tracks the loss and δ as communication rounds increased under two mechanisms. It shows that RR-LADP allocated the privacy budget more carefully, while the loss of global model drops as similar to CDP.

TABLE I: Accuracy and time spent in each round of RR-LADP and CDP. ($\varepsilon = 8$,$\sigma = 1$)

Clients	Rounds	CDP	RR-LADP
100	100	0.73 | 25.9s	**0.80** | 26.5s
1000	200	0.91 | 85.7s	**0.92** | 89.2s
10000	400	0.96 | 315.9s	**0.97** | 336.1s

Table I shows the average accuracy of multiple training and training time per round. RR-LADP achieves higher accuracy than CDP, while increasing the time cost. The accuracy of RR-LADP depends on the effects of RR and LADP, which has been described in the separate experiments. For CDP, the noise addition operation for privacy appears only once on the server side in each training round. However for RR-LADP, all clients in one training round should perform the noise addition operation, bringing the more time delay.

Factors affecting RR-LADP: The accuracy of the final model is affected by multiple factors, such as the batch size, learning rate and other common parameters in machine learning. We have not struggled to find the best combination of these parameters, but only discuss some significant factors in RR-LADP, including **number of clients**, **privacy budget**, and **the noise**.

The number of clients determines the training data and determines the possibility of privacy leakage by affecting the sampling probability. The lower privacy loss in each round means more training rounds and higher accuracy. At the same time, the privacy bound, which is set to $\frac{1}{|K\mathcal{B}|}$, decreases significantly if the number of clients increases.

Privacy budget ε plays a pivotal role in RR-LADP (see Fig. 5). It controls the training process of in two ways. First, it determines the response probability p, which affects the disturbance in the RR mechanism. In fact, $\frac{K^{'}}{K} * p$ in equation 3 represents the part selected by the server to participate in training. The server infer the local updates of a particular client easily if overlap ratio is high. Second, after updating global model in each round, we track δ with the fixed ε. The lower ε, the higher δ, which means it is easier to reach the boundary and the fewer rounds are allowed.

Our mechanism allows to control model performance by choosing the value of σ. The level of noise added in the local updates can directly affect the accuracy of the model. As shown in Fig. 6, the independent variable is the noise parameter σ. When less noise is added, the privacy loss in each round increases. After few rounds of training, δ reaches bound when accuracy of the model is low.

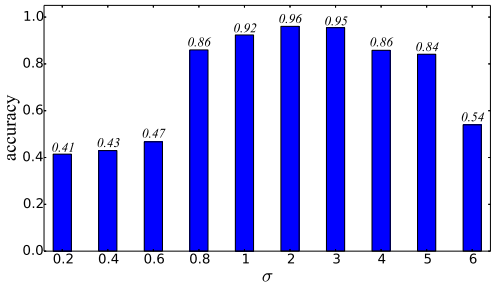

Fig. 6: Results on the accuracy for different noise parameter. ($\varepsilon = 8, K = 1000, K^{'} = 100$)

By adding more noise, the model gains more training rounds with a fixed privacy budget. However, much noise will reduce data availability, and limit the accuracy.

VI. Conclusions

This paper proposes RR-LADP, a federated learning mechanism for IoE, based on randomized response client selection and differentially private client model training. Different from existing approaches that require a trusted server to take charge of the privacy-enhanced process, the core strategy of our approach is to enhance the clients privacy locally to adapt the approach into an environment without trusted servers. It does so by preventing the server from knowing which clients' updates are collected in each round, as well as adding noise adaptively to clients' local updates before submitting them to the server. We show the reliable performance of RR-LADP through experiments with different parameter settings. While providing a higher level privacy-preserving capability, our approach achieves 0.97 training accuracy in the experiments on MNIST. Additionally, as a modified version of traditional federated learning framework, our approach has a potential to be used to train various machine learning models, instead of a single structural model. In the current format of RR-LADP, a global privacy budget is introduced to control the whole training process, with RR focusing on preserving the privacy of a client set and LADP focusing on preserving the private data of each client. Therefore, while the objections to be protected are different, the two mechanisms are sharing a privacy budget. Potential improvements can be achieved by revising such structure. Hence, a future study will give consideration to explore a more delicate allocation of privacy budgets by tracking privacy losses in the two mechanisms separately. In addition, we will improve RR-LADP by applying the mechanism into real-world IoE environment, such as intelligent wearable devices and Internet of Vehicles.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (2017YFB0802204), Key-Area Research and Development Program for Guangdong Province, China (2019B010136001), Basic Research Project of Shenzhen, China (JCYJ20190806143418198), and Basic Research Project of Shenzhen, China (JCYJ20190806142601687). Corresponding authors: Weizhe Zhang and Yang Liu.

Zerui Li is currently a MSc student with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. His research interests include information security and privacy. Contact him at 18S151552@stu.hit.edu.cn.

Yuchen Tian is currently a MSc student with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. His research interests include information security and privacy. Contact him at 19S051060@stu.hit.edu.cn.

Weizhe Zhang is currently a professor in the School of Computer Science and Technology at Harbin Institute of Technology, China. He has published more than 100 academic papers in journals, books, and conference proceedings. He is a senior member of the IEEE. He is the corresponding author of this article. Contact him at wzzhang@hit.edu.cn.

Qing Liao is currently an associate professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. She received her Ph.D degree from the Hong Kong University of Science and Technology. Contact her at liaoqing@hit.edu.cn.

Yang Liu is currently an assistant professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. He received his D.Phil (Ph.D) degree in department of computer science from University of Oxford. He is the corresponding author of this article. Contact him at liu.yang@hit.edu.cn.

Xiaojiang Du is a tenured Full Professor and the Director of the Security And Networking (SAN) Lab in the Department of Computer and Information Sciences at Temple University, Philadelphia, USA. He has authored over 400 journal and conference papers in these areas, as well as a book published by Springer. He is an IEEE Fellow and a Life Member of ACM. Contact him at dxj@ieee.org.

Mohsen Guizani is currently a Professor with the CSE Department, Qatar University, Qatar. He is currently the Editor-in-Chief of the IEEE Network Magazine, serves on the editorial boards of several international technical journals, and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing (Wiley). Contact him at mguizani@ieee.org.