Local Differential Privacy-Based Federated Learning for Internet of Things

Yang Zhao, *Graduate Student Member, IEEE*, Jun Zhao, *Member, IEEE*, Mengmeng Yang, *Member, IEEE*, Teng Wang, *Member, IEEE*, Ning Wang, *Member, IEEE*, Lingjuan Lyu, *Member, IEEE*, Dusit Niyato, *Fellow, IEEE*, and Kwok-Yan Lam, *Senior Member, IEEE*

***Abstract*—The Internet of Vehicles (IoV) is a promising branch of the Internet of Things. IoV simulates a large variety of crowdsourcing applications, such as Waze, Uber, and Amazon Mechanical Turk, etc. Users of these applications report the real-time traffic information to the cloud server which trains a machine learning model based on traffic information reported by users for intelligent traffic management. However, crowdsourcing application owners can easily infer users' location information, traffic information, motor vehicle information, environmental information, etc., which raises severe sensitive personal information privacy concerns of the users. In addition, as the number of vehicles increases, the frequent communication between vehicles and the cloud server incurs unexpected amount of communication cost. To avoid the privacy threat and reduce the communication cost, in this article, we propose to integrate federated learning and local differential privacy (LDP) to facilitate the crowdsourcing applications to achieve the machine learning model. Specifically, we propose four LDP mechanisms to perturb gradients generated by vehicles. The proposed `Three-Outputs` mechanism introduces three different output possibilities to deliver a high accuracy when the privacy budget is small. The output possibilities of `Three-Outputs` can be encoded with two bits to reduce the communication cost. Besides, to maximize the performance when the privacy budget is large, an optimal piecewise mechanism (`PM-OPT`) is proposed. We further propose a suboptimal mechanism (`PM-SUB`) with a simple formula and comparable utility to `PM-OPT`. Then, we build a novel hybrid mechanism by combining `Three-Outputs` and `PM-SUB`. Finally, an `LDP-FedSGD` algorithm is proposed to coordinate the cloud server and vehicles to train the model collaboratively. Extensive experimental results on real-world data sets validate that our proposed algorithms are capable of protecting privacy while guaranteeing utility.**

***Index Terms*—Federated learning, Internet of Things, local differential privacy.**

Manuscript received April 19, 2020; revised July 5, 2020 and August 30, 2020; accepted September 22, 2020. Date of publication November 10, 2020; date of current version May 21, 2021. The work of Yang Zhao and Jun Zhao was supported in part by the Nanyang Technological University (NTU) Startup Grant; in part by the Alibaba-NTU Singapore Joint Research Institute (JRI); in part by the Singapore Ministry of Education Academic Research Fund under Grant Tier 1 RG128/18, Grant Tier 1 RG115/19, Grant Tier 1 RT07/19, Grant Tier 1 RT01/19, and Grant Tier 2 MOE2019-T2-1-176; in part by the NTU-WASP Joint Project; in part by the Singapore National Research Foundation (NRF) under its Strategic Capability Research Centres Funding Initiative: Strategic Centre for Research in Privacy-Preserving Technologies and Systems; in part by the Energy Research Institute @NTU; in part by the Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE under Grant DeST-SCI2019-0012; in part by the AI Singapore 100 Experiments Programme; and in part by the NTU Project for Large Vertical Take-Off and Landing Research Platform. The work of Mengmeng Yang and Kwok-Yan Lam was supported by the National Research Foundation, Singapore under its Strategic Capability Research Centres Funding Initiative. The work of Ning Wang was supported in part by the National Natural Science Foundation of China under 61902365 and in part by the China Postdoctoral Science Foundation under Grant 2019M652473. The work of Dusit Niyato was supported in part by NRF, Singapore, under Singapore Energy Market Authority, Energy Resilience, under Grant NRF2017EWT-EP003-041 and Grant Singapore NRF2015-NRF-ISF001-2277; in part by the Singapore NRF National Satellite of Excellence, Design Science and Technology for Secure Critical Infrastructure NSoE under Grant DeST-SCI2019-0007; in part by the A*STAR-NTU-SUTD Joint Research Grant on Artificial Intelligence for the Future of Manufacturing under Grant RGANS1906; in part by the Wallenberg AI, Autonomous Systems and Software Program and Nanyang Technological University under Grant M4082187 (4080); in part by Singapore Ministry of Education (MOE) Tier 1 under Grant RG16/20; in part by the Alibaba Group through Alibaba Innovative Research Program; and in part by Alibaba-NTU Singapore JRI. *(Corresponding author: Jun Zhao.)*

Yang Zhao, Jun Zhao, Mengmeng Yang, Dusit Niyato, and Kwok-Yan Lam are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: s180049@e.ntu.edu.sg; junzhao@ntu.edu.sg; melody.yang@ntu.edu.sg; dniyato@ntu.edu.sg; kwokyan.lam@ntu.edu.sg).

Teng Wang is with the School of Cyberspace Security, Xi'an University of Posts Telecommunications, Xi'an 710121, China (e-mail: wangteng@xupt.edu.cn).

Ning Wang is with the College of Information Science and Engineering, Ocean University of China, Qingdao 266102, China (e-mail: wangning8687@ouc.edu.cn).

Lingjuan Lyu is with the Department of Computer Science, National University of Singapore, Singapore (e-mail: lingjuanlvsmile@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2020.3037194, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2020.3037194

I. Introduction

THE DEVELOPMENT of sensors and communication technologies for Internet of Things (IoT) have enabled a fast and large-scale collection of user data, which has bred new services and applications, such as the Waze application that provides the intelligent transportation routing service. This kind of service benefits users' daily life, but it may raise privacy concerns of sensitive data, such as users' location information. To address these concerns, we propose a hybrid approach that integrates federated learning (FL) [1] with local differential privacy (LDP) [2] techniques. FL can facilitate the collaborative learning with uploaded gradients from users instead of sharing users' raw data. An honest-but-curious aggregator may be able to leverage users' uploaded gradients to infer the original data [3], [4]. Thus, we deploy LDP noises to gradients to ensure privacy while not compromising the utility of gradients.

Federated Learning With LDP: In addition to LDP mechanisms, FL also provides privacy protection to the data by enabling users to maintain data locally. `FedSGD` algorithm [5]

allows users to submit gradients instead of true data. However, attackers may reverse the gradients to infer original data. By adding LDP noises to the gradients before uploading, we obtain the LDP-based federated stochastic gradient descent `LDP-FedSGD` algorithm, which prevents attackers from deducing original data even though they obtain perturbed gradients. As a result, the FL server gathers and averages users' submitted perturbed gradients to obtain the averaged result to update the global model's parameters.

Existing LDP Mechanisms: Since the proposal of LDP in [7], various LDP mechanisms have been proposed in the literature. Mechanisms for categorical data are presented in [9]–[11]. For numerical data, for which new LDP mechanisms are developed in this article, prior mechanisms of [6]–[8] are discussed as follows. For simplicity, we consider data with a single numeric attribute which has a domain $[-1, 1]$ (after normalization if the original domain is not $[-1, 1]$). Extensions to the case of multiple numeric attributes will be discussed later in this article.

1) *`Laplace` of [7]:* LDP can be understood as a variant of DP, where the difference is the definition of "neighboring data(sets)." In DP, two data sets are neighboring if they differ in just one record; in LDP, any two instances of the user's data are neighboring. Due to this connection between DP and LDP, the classical Laplace mechanism (referred to `Laplace` hereinafter) for DP can thus also be used to achieve LDP. Yet, `Laplace` may not achieve a high utility for some ϵ since ① `Laplace` does not consider the difference between DP and LDP.
2) *`Duchi` of [6]:* In view of the above drawback ① of `Laplace`, Duchi *et al.* [6] introduced alternative mechanisms for LDP. For a single numeric attribute, one mechanism, hereinafter referred to `Duchi` of [6], flips a coin with two possibilities to generate an output, where the probability of each possibility depends on the input. A disadvantage of `Duchi` is as follows: ② Since the output of `Duchi` has only two possibilities, the utility may not be high for large ϵ (intuitively, for large ϵ, the privacy protection is weak so the output should be close to the input which means the output should have many possibilities since the input can take any value in $[-1, 1]$).[1]
3) *Piecewise Mechanism (PM) of [8]:* Due to the drawback ① of `Laplace`, and the drawback ② of `Duchi` above, Wang *et al.* [8] proposed the `PM` which achieves higher utility than `Duchi` for a large ϵ, since the output range of `PM` is continuous and has infinite possibilities, instead of just two possibilities as in `Duchi`. In `PM`, the plot of the output's probability density function with respect to the output value consists of three "pieces," among which the center piece has a higher probability than the other two. As the input increases, the length of the center piece remains unchanged, but the length of the leftmost (resp., rightmost) piece increases (resp., decreases). Since `PM` is tailored for LDP, unlike `Laplace` for LDP, `PM` has a strictly lower worst case variance (i.e., the maximum variance with respect to the input given ϵ) than `Laplace` for *any* ϵ.

[1]Note that any algorithm satisfying DP or LDP has the following property: the set of possible values for the output does not depend on the input (though the output distribution depends on the input). This can be easily seen by contradiction. Suppose an output y is possible for input x but not for x' (x and x' satisfy the neighboring relation in DP or LDP). Then $\mathbb{P}[y|x] > 0$ and $\mathbb{P}[y|x'] = 0$, resulting in $\mathbb{P}[y|x] > e^{\epsilon}\mathbb{P}[y|x']$ and hence violating the privacy requirement ($\mathbb{P}[\cdot|\cdot]$ denotes conditional probability).

TABLE I
COMPARISON OF THE WORST CASE VARIANCES OF EXISTING ϵ-LDP MECHANISMS ON A SINGLE NUMERIC ATTRIBUTE WITH A DOMAIN $[-1, 1]$: `DUCHI` OF [6] GENERATING A BINARY OUTPUT, `LAPLACE` OF [7] WITH THE ADDITION OF LAPLACE NOISE, AND THE PM OF [8]. FOR AN LDP MECHANISM $\mathcal{A}$, ITS WORST CASE VARIANCE IS DENOTED BY $V_{\mathcal{A}}$. WE OBTAIN THIS TABLE BASED ON RESULTS OF [8]

Range of ϵ	Comparison of mechanisms
$0 < \epsilon < 1.29$	$V_{\texttt{Duchi}} < V_{\texttt{PM}} < V_{\texttt{Laplace}}$
$1.29 < \epsilon < 2.32$	$V_{\texttt{PM}} < V_{\texttt{Duchi}} < V_{\texttt{Laplace}}$
$\epsilon > 2.32$	$V_{\texttt{PM}} < V_{\texttt{Laplace}} < V_{\texttt{Duchi}}$

Proposing New LDP Mechanisms: Based on results of [8], Table I on Page 8837 shows that among the three mechanisms `Duchi`, `Laplace`, and `PM`, in terms of the worst case variance, `Duchi` is the best for $0 < \epsilon < 1.29$, while `PM` is the best for $\epsilon > 1.29$. Then a natural research question is that can we propose better or even optimal LDP mechanisms? The optimal LDP mechanism for numeric data is still open in the literature, but the optimal LDP mechanism for categorical data has been discussed by Kairouz *et al.* [12]. In particular, [12] shows that for a categorical attribute (with a limited number of discrete values), the binary and randomized response mechanisms, are universally optimal for very small and large ϵ, respectively. Although [12] handles categorical data, its results can provide the following insight even for continuous numeric data: for very small ϵ, a mechanism generating a binary output should be optimal; for very large ϵ, the optimal mechanism's output should have infinite possibilities. This is also in consistent with the results of Table I on Page 8837. Based on the above insight, intuitively, there may exist a range of medium ϵ where a mechanism with three, or four, or five,..., output possibilities can be optimal. To this end, our first motivation of proposing new LDP mechanisms is to develop a mechanism with three output possibilities such that the mechanism outperforms existing mechanisms of Table I for some ϵ. The outcome of the above motivation is our mechanism called `Three-Outputs`. Since the analysis of `Three-Outputs` is already very complex, we do not consider a mechanism with four, or five,..., output possibilities.

In addition, our second motivation of proposing new LDP mechanisms is that despite the elegance of the `PM` of [8], we can derive the optimal mechanism `PM-OPT` under the "piecewise framework" of [8], in order to improve `PM`. Note that `PM-OPT` is just optimal under the above framework and may not be the optimal LDP mechanism. Since the expressions for `PM-OPT` are quite complex, we present `PM-SUB` which compared with `PM-OPT` is suboptimal, but has simpler expressions and achieves a comparable utility.

Table II on Page 8838 gives a comparison of our and existing LDP mechanisms. For simplicity, we do not include `Laplace` in the comparison since `Laplace` is worse than

TABLE II
COMPARISON OF THE WORST CASE VARIANCES OF OUR AND EXISTING ϵ-LDP MECHANISMS ON A SINGLE NUMERIC ATTRIBUTE WITH A DOMAIN $[-1, 1]$. THREE-OUTPUTS AND PM-SUB ARE OUR MAIN LDP MECHANISMS PROPOSED IN THIS ARTICLE. THE RESULTS IN THIS TABLE SHOW THE ADVANTAGES OF OUR MECHANISMS OVER EXISTING MECHANISMS FOR A WIDE RANGE OF PRIVACY PARAMETER ϵ

Range of ϵ	Comparison of mechanisms
$0 < \epsilon < \ln 2 \approx 0.69$	$V_{\texttt{Duchi}} = V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}}$
$\ln 2 < \epsilon < 1.19$	$V_{\texttt{Three-Outputs}} < V_{\texttt{Duchi}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}}$
$1.19 < \epsilon < 1.29$	$V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{Duchi}} < V_{\texttt{PM}}$
$1.29 < \epsilon < 2.56$	$V_{\texttt{Three-Outputs}} < V_{\texttt{PM-SUB}} < V_{\texttt{PM}} < V_{\texttt{Duchi}}$
$2.56 < \epsilon < 3.27$	$V_{\texttt{PM-SUB}} < V_{\texttt{Three-Outputs}} < V_{\texttt{PM}} < V_{\texttt{Duchi}}$
$\epsilon > 3.27$	$V_{\texttt{PM-SUB}} < V_{\texttt{PM}} < V_{\texttt{Three-Outputs}} < V_{\texttt{Duchi}}$

PM for any ϵ according to Table I. As shown in Table II, in terms of the worst case variance for a single numeric attribute with a domain $[-1, 1]$, Three-Outputs outperforms Duchi for $\epsilon > \ln 2 \approx 0.69$ (and is the same as Duchi for $\epsilon \leq \ln 2$), while PM-SUB beats PM for any ϵ; moreover, Three-Outputs outperforms both Duchi and PM for $\ln 2 < \epsilon < 3.27$, while PM-SUB beats both Duchi and PM for $\epsilon > 1.19$.

We also follow the practice of [8], which combines different mechanisms Duchi and PM to obtain a hybrid mechanism HM. HM has a lower worst case variance than those of Duchi and PM. In this article, we combine Three-Outputs and PM-SUB to propose HM-TP. The intuition is as follows. Given ϵ, the variances of Three-Outputs and PM-SUB [denoted by $T_\epsilon(x)$ and $P_\epsilon(x)$] depend on the input x, and their maximal values (i.e., the worst case variances) may be taken at different x. Hence, for a hybrid mechanism which probabilistically uses Three-Outputs with probability q or PM-SUB with probability $1 - q$, given ϵ, the worst case variance $\max_x(q \cdot T_\epsilon(x) + (1 - q) \cdot P_\epsilon(x))$ may be strictly smaller than the minimum of $\max_x T_\epsilon(x)$ and $\max_x P_\epsilon(x)$. We optimize q for each ϵ to obtain HM-TP. Due to the already complex expression of PM-OPT, we do not consider the combination of PM-OPT and Three-Outputs.

Contributions: Our contributions can be summarized as follows.

1) Using the LDP-FedSGD algorithm for FL in Internet of Vehicles (IoV) as a motivating context, we present novel LDP mechanisms for numeric data with a continuous domain. Among our proposed mechanisms, Three-Outputs and PM-SUB outperform existing mechanisms for a wide range of ϵ, as shown in the theoretical results in Table II and confirmed by experiments. In terms of comparing our Three-Outputs and PM-SUB, we have: Three-Outputs, whose output has three possibilities, is better for small ϵ, while PM-SUB, whose output can take infinite possibilities of an interval, has higher utility for large ϵ. Our PM-SUB is a slightly suboptimal version of our PM-OPT to simplify the expressions. We further combine Three-Outputs and PM-SUB to obtain a hybrid mechanism HM-TP, which achieves even higher utility.
2) We discretize the continuous output ranges of our proposed mechanisms PM-SUB and PM-OPT. Through the discretization postprocessing, we enable vehicles to use our proposed mechanisms. In Section VIII, we confirm that the discretization postprocessing algorithm maintains utility with our experiments, while reducing the communication cost.
3) Experimental evaluation of our proposed mechanisms on real-world data sets and synthetic data sets demonstrates that our proposed mechanisms achieve higher accuracy in estimating the mean frequency of the data and performing empirical risk minimization tasks than existing approaches.

Organization: In the following, Section II introduces the preliminaries. Then, we introduce related works in Section III. Then, we illustrate the system model and the LDP-based FedSGD (LDP-FedSGD) algorithm in Section IV. Section V presents the problem formation. Section VI proposes novel solutions for the single numerical data estimation. Section VII illustrates proposed mechanisms used for multidimensional numerical data estimation. Section VIII demonstrates our experimental results. Section IX concludes this article.

II. PRELIMINARIES

In LDP, users complete the perturbation by themselves. To protect users' privacy, each user runs a random perturbation algorithm $\mathcal{M}$, and then he sends perturbed results to the aggregator. The privacy budget ϵ controls the privacy-utility tradeoff, and a higher privacy budget means a lower privacy protection. As a result of this, we define LDP as follows.

Definition 1 (LDP): Let $\mathcal{M}$ be a randomized function with domain $\mathbb{X}$ and range $\mathbb{Y}$; i.e., $\mathcal{M}$ maps each element in $\mathbb{X}$ to a probability distribution with sample space $\mathbb{Y}$. For a nonnegative ϵ, the randomized mechanism $\mathcal{M}$ satisfies ϵ-LDP if

$$\left| \ln \frac{\mathbb{P}_{\mathcal{M}}[Y \in S|x]}{\mathbb{P}_{\mathcal{M}}[Y \in S|x']} \right| \leq \epsilon, \ \forall x, x' \in \mathbb{X}, \ \forall S \subseteq \mathbb{Y}. \tag{1}$$

$\mathbb{P}_{\mathcal{M}}[\cdot|\cdot]$ means conditional probability distribution depending on $\mathcal{M}$. In LDP, the random perturbation is performed by users instead of a centralized aggregator. Centralized aggregator only receives perturbed results which make sure that the aggregator is unable to distinguish whether the true tuple is x or x' with high confidence (controlled by the privacy budget ϵ).

III. RELATED WORK

Recently, LDP has attracted much attention [13]–[20]. Several mechanisms for numeric data estimation have been proposed [6]–[8], [21].

1) Dwork *et al.* [7] proposed the Laplace mechanism which adds the Laplace noise to real 1-D data directly. The Laplace mechanism is originally used in the centralized differential privacy mechanism, and it can be applied to LDP directly.
2) For a single numeric attribute with a domain $[-1, 1]$, Duchi *et al.* [6] proposed an LDP framework that provides output from $\{-C, C\}$, where $C > 1$.
3) Wang *et al.* [8] proposed the PM which offers an output that contains infinite possibilities in the range of $[-A, A]$, where $A > 1$. In addition, they apply LDP

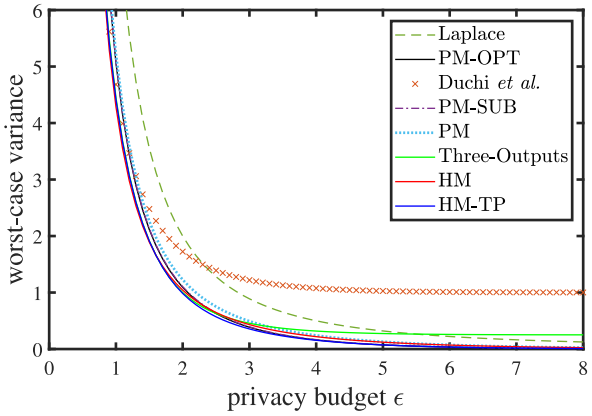


Fig. 1. Different mechanisms' worst case noise variance for 1-D numeric data versus the privacy budget ϵ.

mechanism to preserve the privacy of gradients generated during machine learning tasks. Both approaches by Duchi *et al.* [6] and Wang *et al.* [8] can be extended to the case of multidimensional numerical data.

Deficiencies of Existing Solutions: Fig. 1 illustrates that when $\epsilon \leq 2.3$, Laplace mechanism's worst case noise variance is larger than that of Duchi *et al.*'s [22] solution; however, the Laplace mechanism outperforms Duchi *et al.*'s [22] solution if ϵ is larger. The worst case noise variance in `PM` is smaller than that of Laplace and Duchi *et al.*'s [22] solution when ϵ is large. The `HM` mechanism outperforms other existing solutions by taking advantage of Duchi *et al.*'s [22] solution when ϵ is small and `PM` when ϵ is large. However, `PM` and `HM`'s outputs have infinite possibilities that are hard to encode. We would like to find a mechanism that can improve the utility of existing mechanisms. In addition, we believe there is a mechanism that retains a high utility and is easy to encode its outputs. Based on the above intuition, we propose four novel mechanisms that can be used by vehicles in Section VI.

In addition, LDP has been widely used in the research of IoT [23]–[27]. For example, Xu *et al.* [23] integrate deep learning with LDP techniques and apply them to protect users' privacy in edge computing. They develop an EdgeSanitizer framework that forms a new protection layer against sensitive inference by leveraging a deep learning model to mask the learned features with noise and minimize data. Choi *et al.* [24] explore the feasibility of applying LDP on ultralow-power (ULP) systems. They use resampling, thresholding, and a privacy budget control algorithm to overcome the low resolution and fixed point nature of ULPs. He *et al.* [25] address the location privacy and usage pattern privacy induced by the mobile edge computing's wireless task offloading feature by proposing a privacy-aware task offloading scheduling algorithm based on constrained Markov decision process. Li *et al.* [26] propose a scheme for privacy-preserving data aggregation in the mobile edge computing to assist IoT applications with three participants, i.e., a public cloud center (PCC), an edge server (ES), and a terminal device (TD). TDs generate and encrypt data and send them to the ES, and then the ES submits the aggregated data to the PCC. The PCC uses its private key to recover the aggregated plaintext data. Their scheme provides source authentication and integrity and guarantees the data privacy of the TDs. In addition, their scheme can save half of the communication cost. To protect the privacy of massive data generated from IoT platforms, Arachchige *et al.* [27] design an LDP mechanism named as LATENT for deep learning. A randomization layer between the convolutional module and the fully connected module is added to the LATENT to perturb data before data leave data owners for machine learning services. Pihur [28] propose the Podium Mechanism which is similar to our `PM-SUB`, but his mechanism is applicable to DP instead of LDP.

Moreover, FL or collaborative learning is an emerging distributed machine learning paradigm, and it is widely used to address data privacy problem in machine learning [1], [29]. Recently, FL is explored extensively in the IoT recently [30]–[34]. Lim *et al.* [30] survey FL applications in mobile edge network comprehensively, including algorithms, applications and potential research problems, etc. Besides, Lu *et al.* [32] propose CLONE which is a collaborative learning framework on the edges for connected vehicles, and it reduces the training time while guaranteeing the prediction accuracy. Different from CLONE, our proposed approach utilizes LDP noises to protect the privacy of the uploaded data. Furthermore, Fantacci and Picano [33] leverage FL to protect the privacy of mobile edge computing, while Saputra *et al.* [34] apply FL to predict the energy demand for electrical vehicle networks.

Furthermore, there have been many papers on FL and differential privacy, such as [31], [35]–[43]. For example, Truex *et al.* [36] utilize both secure multiparty computation and centralized differential privacy to prevent inference over both the messages exchanged in the process of training the model. However, they do not analyze the impact of the privacy budget on performance of FL. Hu *et al.* [37] propose a privacy-preserving FL approach for learning effective personalized models. They use Gaussian mechanism, a centralized DP mechanism, to protect the privacy of the model. Compared with them, our proposed LDP mechanisms provide a stronger privacy protection using the LDP mechanism. Hao *et al.* [31] propose a differential enhanced FL scheme for industrial artificial industry. Triastcyn and Faltings [39] employ Bayesian differential privacy on FL. They make use of the centralized differential privacy mechanism to protect the privacy of gradients, but we leverage a stronger privacy-preserving mechanism (LDP) to protect each vehicle's privacy.

Additionally, DP can be applied to various FL algorithms, such as `FedSGD` [5] and `FedAvg` [1]. `FedAvg` requires users to upload model parameters instead of gradients in `FedSGD`. The advantage of `FedAvg` is that it allows users to train the model for multiple rounds locally before submitting gradients. Brendan *et al.* [35] propose to apply centralized DP to `FedAvg` and `FedSGD` algorithm. In our paper, we deploy LDP mechanisms to gradients in `FedSGD` algorithm. Our future work is to develop LDP mechanisms for state-of-the-art FL algorithms. Zhao *et al.* [44] propose a SecProbe mechanism to protect privacy and quality of participants' data by leveraging exponential mechanism and functional mechanism of differential privacy. SecProbe guarantees the high accuracy

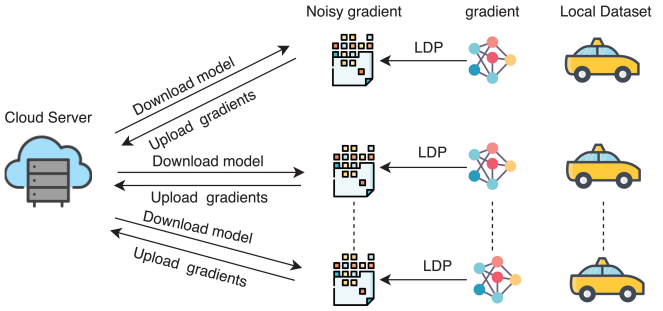


Fig. 2. System design.

as well as privacy protection. In addition, it prevents unreliable participants in the collaborative learning.

IV. System Model and Local Differential Privacy-Based FedSGD Algorithm

A. System Model

In this work, we consider a scenario where a number of vehicles are connected with a cloud server as Fig. 2. Each vehicle is responsible for continuously performing training and inference locally based on data that it collects and the model initiated by the cloud server. The local training dataset is never uploaded to the cloud server. After finishing predefined epochs locally, the cloud server calculates the average of uploaded gradients from vehicles and updates the global model with the average. The FL aggregator is honest-but-curious or semi-honest, which follows the FL protocol but it will try to learn additional information using received data [36], [45]. With the injected LDP noise, servers or attackers cannot retrieve users' information by reversing their uploaded gradients [3], [4]. Thus, there is a need to deploy LDP mechanisms to the FL to develop a communication-efficient LDP-based FL algorithm.

B. Federated Learning With LDP: LDP-FedSGD

In addition, we propose (`LDP-FedSGD`) for our proposed system. Details of `LDP-FedSGD` are given in Algorithm 1. Unlike the FedAvg algorithm, in the `FedSGD` algorithm, clients (i.e., vehicles) upload updated gradients instead of model parameters to the central aggregator (i.e., cloud server) [5]. However, compared with the standard `FedSGD` [5], we add our proposed LDP mechanism proposed in Section VII to prevent the privacy leakage of gradients. Each vehicle locally takes one step of gradient descent on the current model using its local data, and then it perturbs the true gradient with Algorithm 6. The server aggregates and averages the updated gradients from vehicles and then updates the model. To reduce the communication rounds, we separate vehicles into groups, so that the cloud server updates the model after gathering gradient updates from vehicles in a group. In the following sections, we will introduce how we obtain the LDP algorithm in detail.

C. Comparing LDP-FedSGD With Other Privacy-Preserving Federated Learning Paradigms

The `LDP-FedSGD` algorithm incorporates LDP into FL. In addition to `LDP-FedSGD`, one may be interested in other

Algorithm 1: LDP-FedSGD Algorithm

Server executes:
Server initializes the parameter as θ_0;
for *t from* 1 *to maximal iteration number* **do**
  Server sends θ_{t-1} to vehicles in group G_t;
  for *each vehicle i in Group* G_t **do**
    VehicleUpdate(i, ΔL):
  Server computes the average of the noisy gradient of group G_t and updates the parameter from θ_{t-1} to θ_t: $\theta_t \leftarrow \theta_{t-1} - \eta \cdot 1/|G_t| \sum_{i \in G_t} \mathcal{M}(\Delta L(\theta_{t-1}; x_i))$, where η_t is the learning rate;
  if θ_t *and* θ_{t-1} *are close enough or these remains no vehicle which has not participated in the computation* **then**
    break;
  $t \rightarrow t+1$;
VehicleUpdate (i, ΔL)**:**
Compute the (true) gradient $\Delta L(\theta_{t-1}; x_i)$, where x_i is vehicle i's data;
Use LDP-compliant algorithm $\mathcal{M}$ to compute the noisy gradient $\mathcal{M}(\Delta(\theta_{t-1}; x_i))$;

ways of using DP in FL. To explain them, we categorize combinations of DP and FL (or distributed computations in general) by considering the place of perturbation (distributed/centralized perturbation) and privacy granularity (user-level/record-level privacy protection) [46].

1) *Distributed/Centralized Perturbation:* Note that differential privacy is achieved by introducing perturbation. Distributed perturbation considers an honest-but-curious aggregator, while centralized perturbation needs a trusted aggregator. Both perturbation methods defend against external inference attacks after model publishing.
2) *User-Level/Record-Level Privacy Protection:* In general, a differentially private algorithm ensures that the probability distributions of the outputs on two neighboring data sets do not differ much. The distinction between user-level and record-level privacy protection lies in how neighboring data sets are defined. We define that two data sets are user-neighboring if one data set can be formed from the other data set by adding or removing all one user's records arbitrarily. We define that two data sets are record-neighboring if one data set can be obtained from the other data set by changing a *single* record of one user.

Based on the above, we further obtain four paradigms as follows: 1) user-level privacy protection with distributed perturbation (ULDP); 2) record-level privacy protection with distributed perturbation (RLDP); 3) record-level privacy protection with centralized perturbation (RLCP); and 4) user-level privacy protection with centralized perturbation (ULCP). The details are as follows and can also be found in the second author's prior work [46].

1) In ULDP, each user i selects a privacy parameter ϵ_i and applies a randomization algorithm Y_i such that given any

two instances x_i and x_i' of user i's data (which are *user*-neighboring), and for any possible subset of outputs[2] $\mathcal{Y}_i$ of Y_i, we obtain $\mathbb{P}[Y_i \in \mathcal{Y}_i|x_i] \leq e^{\epsilon_i} \times \mathbb{P}\big[Y_i \in \mathcal{Y}_i|x_i'\big]$. Clearly, ULDP is achieved by each user implementing LDP studied in this article.

2) In RLDP, each user i selects a privacy parameter ϵ_i and applies a randomization algorithm Y_i such that for any two *record*-neighboring instances x_i and x_i' of user i's data (i.e., x_i and x_i' differ in only one record), and for any possible subset of outputs $\mathcal{Y}_i$ of Y_i, we have $\mathbb{P}[Y_i \in \mathcal{Y}_i|x_i] \leq e^{\epsilon_i} \times \mathbb{P}\big[Y_i \in \mathcal{Y}_i|x_i'\big]$. Clearly, in RLDP, what each user does is just to apply standard differential privacy. In contrast, in ULDP above, each user applies LDP.
3) In ϵ-RLCP, the aggregator sets a privacy parameter ϵ and applies a randomization algorithm Y such that for any user i, for any two *record*-neighboring instances x_i and x_i' of user i's data, and for any possible subset of outputs[2] $\mathcal{Y}$ of Y, we obtain $\mathbb{P}[Y \in \mathcal{Y}|x_i] \leq e^{\epsilon} \times \mathbb{P}\big[Y \in \mathcal{Y}|x_i'\big]$. In other words, in RLCP, the aggregator applies standard differential privacy. For the aggregator to implement RLCP well, typically the aggregator should be able to bound the impact of each record on the information sent from a user to the aggregator. Further discussions on this can be interesting, but we do not present more details since RLCP is not our paper's focus.
4) In ϵ-ULCP, the aggregator sets a privacy parameter ϵ and applies a randomization algorithm Y so that for any user i, for any two instances x_i and x_i' of user i's data (which are *user*-neighboring), and for any possible subset of outputs $\mathcal{Y}$ of Y, we have $\mathbb{P}[Y \in \mathcal{Y}|x_i] \leq e^{\epsilon} \times \mathbb{P}\big[Y \in \mathcal{Y}|x_i'\big]$. The difference RLCP and ULCP is that RLCP achieves record-level privacy protection while ULCP ensures the stronger user-level privacy protection.

In the case of distributed perturbation, when all users set the same privacy parameter ϵ, we refer to ULDP and RLDP above as ϵ-ULDP and ϵ-RLDP, respectively. Table III presents a comparison of ϵ-ULDP, ϵ-RLDP, ϵ-RLCP, and ϵ-ULCP. In this article's focus, each user applies ϵ-LDP, so our framework is under ULDP. The reasons that we consider ULDP instead of RLDP, RLCP, and ULCP are as follows.

1) We do not consider RLDP which implements perturbation at each user via standard differential privacy, since we aim to achieve user-level privacy protection instead of the weaker record-level privacy protection (a vehicle is a user in our IoV applications and may have multiple records). The motivation is that often much data from a vehicle may be about the vehicle's regular driver, and it often makes more sense to protect all data about the regular driver instead of just protecting each single record. A similar argument has been recently stated in [35], which incorporates user-level differential privacy into the training process of FL for language modeling. Specifically, [35] considers user-level privacy to protect the privacy of all typed words of a user, and explains that such privacy protection is more reasonable than protecting individual words as in the case of record-level privacy. In addition, although we can compute the level of user-level privacy from record-level privacy via the group privacy property of differential privacy (see [47, Th. 2.2]), but this may significantly increase the privacy parameter and hence weaken the privacy protection if a user has many records (note that a larger privacy parameter ϵ in ϵ-DP means weaker privacy protection). More specifically, for a user with m records, according to the group privacy property [47], the privacy protection strength for the user under ϵ-record-level privacy is just as that under $m\epsilon$-user-level privacy (i.e., for a user with m records, ϵ-RLDP ensures $m\epsilon$-ULDP; ϵ-RLCP ensures $m\epsilon$-ULCP).
2) We also do not investigate RLCP and ULCP since this article considers a honest-but-curious aggregator instead of a trusted aggregator. The aggregator is not completely trusted, so the perturbation is implemented at each user (i.e., vehicle in IoV).

[2]For simplicity, we slightly abuse the notation and denote the output of algorithm Y_i (resp., algorithm Y) by Y_i (resp., Y).

TABLE III
WE COMPARE DIFFERENT PRIVACY NOTIONS IN THIS TABLE. IN THIS ARTICLE, WE FOCUS ON ϵ-LDP WHICH ACHIEVES ULDP. WE DO NOT CONSIDER RLDP WHICH IMPLEMENTS PERTURBATION AT EACH USER VIA STANDARD DIFFERENTIAL PRIVACY, SINCE WE AIM TO ACHIEVE USER-LEVEL PRIVACY PROTECTION INSTEAD OF THE WEAKER RECORD-LEVEL PRIVACY PROTECTION (A VEHICLE IS A USER IN OUR IOV APPLICATIONS AND MAY HAVE MULTIPLE RECORDS). WE ALSO DO NOT INVESTIGATE RECORD/USER-LEVEL PRIVACY PROTECTION WITH CENTRALIZED PERTURBATION (RLCP/ULCP) SINCE THIS ARTICLE CONSIDERS A HONEST-BUT-CURIOUS AGGREGATOR INSTEAD OF A TRUSTED AGGREGATOR

privacy granularity and place of perturbation	privacy property	adversary model
ϵ-LDP (defined for **distributed** perturbation)	ϵ-ULDP	defend against a honest-but-curious aggregator &external attacks after model publishing
ϵ-DP with **distributed** perturbation	ϵ-RLDP	
ϵ-DP with **centralized** perturbation	ϵ-RLCP	trusted aggregator; defend against external attacks after model publishing
user-level privacy with **centralized** perturbation	ϵ-ULCP	

V. PROBLEM FORMATION

Let x be a user's true value, and Y be the perturbed value. Under the perturbation mechanism $\mathcal{M}$, we use $\mathbb{E}_{\mathcal{M}}[Y|x]$ to denote the expectation of the randomized output Y given input x. $\mathrm{Var}_{\mathcal{M}}[Y|x]$ is the variance of output Y given input x. $\mathrm{MaxVar}(\mathcal{M})$ denotes the worst case $\mathrm{Var}_{\mathcal{M}}[Y|x]$. We are interested in finding a privatization mechanism $\mathcal{M}$ that minimizes $\mathrm{MaxVar}(\mathcal{M})$ by solving the following constraint minimization problem:

$$\begin{aligned} \min_{\mathcal{M}} \quad & \mathrm{MaxVar}(\mathcal{M}) \\ \text{s.t.} \quad & \text{Eq. (1)}, \\ & \mathbb{E}_{\mathcal{M}}[Y|x] = x, \ \text{and} \\ & \mathbb{P}_{\mathcal{M}}[Y \in \mathbb{Y}|x] = 1. \end{aligned}$$

The second constraint illustrates that our estimator is unbiased, and the third constraint shows the proper distribution where

Algorithm 2: `Three-Outputs` Mechanism for 1-D Numeric Data

Input: tuple $x \in [-1, 1]$ and privacy parameter ϵ.
Output: tuple $Y \in \{-C, 0, C\}$.

Sampling a random variable u with the probability distribution as follows:

$$\mathbb{P}[u = -1] = P_{-C\leftarrow x},$$
$$\mathbb{P}[u = 0] = P_{0\leftarrow x}, \text{ and}$$
$$\mathbb{P}[u = 1] = P_{C\leftarrow x},$$

where $P_{-C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{C\leftarrow x}$ are given in (2), (3) and (3).

if $u = -1$ **then**
  $Y = -C$;
else if $u = 0$ **then**
  $Y = 0$;
else
  $Y = C$;
return Y;

$\mathbb{Y}$ is the range of randomized function $\mathcal{M}$. In the following sections, if $\mathcal{M}$ is clear from the context, we omit the subscript $\mathcal{M}$ for simplicity.

VI. Mechanisms for Estimation of Single Numeric Attribute

To solve the problem in Section V, we propose four LDP mechanisms: `Three-Outputs`, `PM-OPT`, `PM-SUB`, and `HM-TP`. Fig. 1 compares the worst case noise variances of existing mechanisms and our proposed mechanisms. `Three-Outputs` has three discrete output possibilities, which incurs little communication cost because two bits are enough to encode three different outputs. Moreover, it achieves a small worst case noise variance in the high privacy regime (small privacy budget ϵ). However, to maintain a low worst case noise variance in the low privacy regime (large privacy budget ϵ), we propose `PM-OPT` and `PM-SUB`. Both of them achieve higher accuracies than `Three-Outputs` and other existing solutions when the privacy budget ϵ is large. Additionally, we discretize their continuous ranges of output for vehicles to encode using a postprocessing discretization algorithm. In the following sections, we will explain our proposed four mechanisms and the postprocessing discretization algorithm in detail respectively.

A. `Three-Outputs` Mechanism

Now, we propose a mechanism with three output possibilities named as `Three-Outputs` which is illustrated in Algorithm 2. `Three-Outputs` ensures low communication cost while achieving a smaller worst case noise variance than existing solutions in the high privacy regime (small privacy budget ϵ). Duchi *et al.*'s [22] solution contains two output possibilities, and it outperforms other approaches when the privacy budget is small. However, Kairouz *et al.* [12] prove that two outputs are not always optimal as ϵ increases. By outputting three values instead of two, `Three-Outputs` improves the performance as the privacy budget increases, which is shown in Fig. 1. When the privacy budget is small, `Three-Outputs` is equivalent to Duchi *et al.*'s [22] solution.

For notional simplicity, given a mechanism $\mathcal{M}$, we often write $\mathbb{P}_{\mathcal{M}}[Y = y|X = x]$ as $P_{y\leftarrow x}(\mathcal{M})$ below. We also sometimes omit $\mathcal{M}$ to obtain $\mathbb{P}[Y = y|X = x]$ and $P_{y\leftarrow x}$.

Given a tuple $x \in [-1, 1]$, `Three-Outputs` returns a perturbed value Y that equals $-C$, 0 or C with probabilities defined by

$$P_{-C\leftarrow x} = \begin{cases} \dfrac{1-P_{0\leftarrow 0}}{2} + \left(\dfrac{1-P_{0\leftarrow 0}}{2} - \dfrac{e^\epsilon - P_{0\leftarrow 0}}{e^\epsilon(e^\epsilon+1)}\right)x, \\ \quad \text{if } 0 \le x \le 1 \\ \dfrac{1-P_{0\leftarrow 0}}{2} + \left(\dfrac{e^\epsilon - P_{0\leftarrow 0}}{e^\epsilon+1} - \dfrac{1-P_{0\leftarrow 0}}{2}\right)x, \\ \quad \text{if } -1 \le x \le 0 \end{cases} \tag{2}$$

$$P_{C\leftarrow x} = \begin{cases} \dfrac{1-P_{0\leftarrow 0}}{2} + \left(\dfrac{e^\epsilon - P_{0\leftarrow 0}}{e^\epsilon+1} - \dfrac{1-P_{0\leftarrow 0}}{2}\right)x, \\ \quad \text{if } 0 \le x \le 1 \\ \dfrac{1-P_{0\leftarrow 0}}{2} + \left(\dfrac{1-P_{0\leftarrow 0}}{2} - \dfrac{e^\epsilon - P_{0\leftarrow 0}}{e^\epsilon(e^\epsilon+1)}\right)x, \\ \quad \text{if } -1 \le x \le 0, \end{cases} \tag{3}$$

and

$$P_{0\leftarrow x} = P_{0\leftarrow 0} + \left(\frac{P_{0\leftarrow 0}}{e^\epsilon} - P_{0\leftarrow 0}\right)x, \text{ if } -1 \le x \le 1 \tag{4}$$

where $P_{0\leftarrow 0}$ is defined by

$$P_{0\leftarrow 0} \coloneqq \begin{cases} 0, \quad \text{if } \epsilon < \ln 2, \\ -\dfrac{1}{6}\left(-e^{2\epsilon} - 4e^\epsilon - 5 \right. \\ \left. + 2\sqrt{\Delta_0}\cos\left(\frac{\pi}{3} + \frac{1}{3}\arccos\left(-\frac{\Delta_1}{2\Delta_0^{\frac{3}{2}}}\right)\right)\right), \\ \quad \text{if } \ln 2 \le \epsilon \le \epsilon' \\ \dfrac{e^\epsilon}{e^\epsilon+2}, \text{ if } \epsilon > \epsilon' \end{cases} \tag{5}$$

in which

$$\Delta_0 \coloneqq e^{4\epsilon} + 14e^{3\epsilon} + 50e^{2\epsilon} - 2e^\epsilon + 25 \tag{6}$$

$$\Delta_1 \coloneqq -2e^{6\epsilon} - 42e^{5\epsilon} - 270e^{4\epsilon} - 404e^{3\epsilon} - 918e^{2\epsilon} + 30e^\epsilon - 250 \tag{7}$$

$$\text{and } \epsilon' \coloneqq \ln\left(\frac{3+\sqrt{65}}{2}\right) \approx \ln 5.53. \tag{8}$$

Next, we will show how we derive the above probabilities. For a mechanism which uses $x \in [-1, 1]$ as the input and only

three possibilities $-C, 0, C$ for the output value, it satisfies

$$\begin{cases} \epsilon\text{-LDP:} \dfrac{P_{C\leftarrow x}}{P_{C\leftarrow x'}}, \dfrac{P_{0\leftarrow x}}{P_{0\leftarrow x'}}, \dfrac{P_{-C\leftarrow x}}{P_{-C\leftarrow x'}} \in [e^{-\epsilon} e^{\epsilon}] & \text{(9a)} \\ \text{unbiased estimation:} \\ \quad C \cdot P_{C\leftarrow x} + 0 \cdot P_{0\leftarrow x} + (-C) \cdot P_{-C\leftarrow x} = x & \text{(9b)} \\ \text{proper distribution:} \\ \quad P_{y\leftarrow x} \geq 0 \text{ and } P_{C\leftarrow x} + P_{0\leftarrow x} + P_{-C\leftarrow x} = 1. & \text{(9c)} \end{cases}$$

To calculate values of $P_{C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{-C\leftarrow x}$, we use Lemma 1 below to convert a mechanism $\mathcal{M}_1$ satisfying the requirements in (9a)–(9c) to a symmetric mechanism $\mathcal{M}_2$. Then, we use Lemma 2 below to transform the symmetric mechanism further to $\mathcal{M}_3$ whose worst case noise variance is smaller than $\mathcal{M}_2$'s. Next, we use $P_{0\leftarrow 1}$ to represent other probabilities, and then we prove that we get the minimum variance when $P_{0\leftarrow 0} = e^{\epsilon} P_{0\leftarrow 1}$ using Lemma 3. Finally, Lemmas 4 and 5 are used to obtain values for $P_{0\leftarrow 0}$ and the worst case noise variance of `Three-Outputs`, respectively. Thus, we can obtain values of $P_{C\leftarrow x}$, $P_{0\leftarrow x}$ and $P_{-C\leftarrow x}$ using $P_{0\leftarrow 0}$. In the following, we will illustrate above processes in detail.

By symmetry, for any $x \in [-1, 1]$, we enforce

$$\begin{cases} P_{C\leftarrow x} = P_{-C\leftarrow -x} & \text{(10a)} \\ P_{0\leftarrow x} = P_{0\leftarrow -x}, & \text{(10b)} \end{cases}$$

where (10b) can be derived from (10a). The formal justification of (10a) and (10b) is given by Lemma 1 below. Since the input domain $[-1, 1]$ is symmetric, we can transform any mechanism satisfying requirements in (9a)–(9c) to a symmetric mechanism while guaranteeing the worst case noise variance will not increase in Lemma 1. Thus, we can derive probabilities when $x \in [-1, 0]$ using probabilities when $x \in [0, 1]$ based on the symmetry.

Lemma 1: For a mechanism $\mathcal{M}_1$ satisfying the requirements in (9a)–(9c), the following symmetrization process to obtain a mechanism $\mathcal{M}_2$ will not increase (i.e., will reduce or not change) the worst case noise variance, while mechanism $\mathcal{M}_2$ still satisfies the requirements in (9a)–(9c). Symmetrization: For $x \in [-1, 1]$

$$\begin{aligned} P_{C\leftarrow x}(\mathcal{M}_2) &= P_{-C\leftarrow -x}(\mathcal{M}_2) \\ &= \frac{P_{C\leftarrow x}(\mathcal{M}_1) + P_{-C\leftarrow -x}(\mathcal{M}_1)}{2} \end{aligned} \tag{11}$$

$$P_{0\leftarrow x}(\mathcal{M}_2) = P_{0\leftarrow -x}(\mathcal{M}_2) = \frac{P_{0\leftarrow x}(\mathcal{M}_1) + P_{0\leftarrow -x}(\mathcal{M}_1)}{2}. \tag{12}$$

Proof: The proof details are given in Appendix A of the supplementary material. ∎

Based on Lemma 1, we define a symmetric mechanism as follows.

Symmetric Mechanism: A mechanism under (9a)–(9c) is called a symmetric mechanism if it satisfies (10a) and (10b). In the following, we only consider the symmetric mechanism $\mathcal{M}_2$.

Now, we design probabilities for the symmetric mechanism $\mathcal{M}_2$. As $\mathcal{M}_2$ satisfies the unbiased estimation which is a linear relationship, we set probabilities as piecewise linear functions of x as follows.

Case 1: For $x \in [0, 1]$

$$P_{C\leftarrow x} = P_{C\leftarrow 0} + (P_{C\leftarrow 1} - P_{C\leftarrow 0})x, \tag{13}$$

$$P_{-C\leftarrow x} = P_{-C\leftarrow 0} - (P_{-C\leftarrow 0} - P_{-C\leftarrow 1})x \tag{14}$$

$$\begin{aligned} P_{0\leftarrow x} &= 1 - P_{-C\leftarrow 0} - P_{C\leftarrow 0} \\ &\quad + (P_{-C\leftarrow 0} - P_{C\leftarrow 0} + P_{-C\leftarrow 1} - P_{C\leftarrow 1})x. \end{aligned} \tag{15}$$

Case 2: For $x \in [-1, 0]$

$$P_{C\leftarrow x} = P_{C\leftarrow 0} + (P_{C\leftarrow 0} - P_{C\leftarrow -1})x \tag{16}$$

$$P_{-C\leftarrow x} = P_{-C\leftarrow 0} - (P_{-C\leftarrow -1} - P_{-C\leftarrow 0})x \tag{17}$$

$$\begin{aligned} P_{0\leftarrow x} &= 1 - P_{-C\leftarrow 0} - P_{C\leftarrow 0} \\ &\quad + (P_{-C\leftarrow 0} - P_{C\leftarrow 0} + P_{-C\leftarrow 1} - P_{C\leftarrow 1})x. \end{aligned} \tag{18}$$

Then, we may assign values to our designed probabilities above. We find that if a symmetric mechanism satisfies (19a) and (19b), it obtains a smaller worst case noise variance. From Lemma 2, we enforce

$$\begin{cases} P_{C\leftarrow 1} = e^{\epsilon} P_{C\leftarrow -1} & \text{(19a)} \\ P_{-C\leftarrow -1} = e^{\epsilon} P_{-C\leftarrow 1}. & \text{(19b)} \end{cases}$$

Hence, given a symmetric mechanism $\mathcal{M}_2$ satisfying Inequality (20), we can transform it to a new symmetric mechanism $\mathcal{M}_3$ which satisfies (19a) and (19b) through processes of (21)–(23) until $P_{C\leftarrow -1} = e^{\epsilon} P_{-C\leftarrow 1}$. After transformation, the new mechanism $\mathcal{M}_3$ achieves a smaller worst case noise variance than mechanism $\mathcal{M}_2$. Therefore, we use the new symmetric mechanism $\mathcal{M}_3$ to replace $\mathcal{M}_2$ in the future's discussion. Details of transformation are in the Lemma 2.

Lemma 2: For a symmetric mechanism $\mathcal{M}_2$, if

$$P_{C\leftarrow 1}(\mathcal{M}_2) < e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2) \tag{20}$$

we set a symmetric mechanism $\mathcal{M}_3$ as follows: For $x \in [-1, 1]$

$$\begin{aligned} P_{C\leftarrow x}(\mathcal{M}_3) &= P_{-C\leftarrow -x}(\mathcal{M}_3) \\ &= P_{C\leftarrow x}(\mathcal{M}_2) \\ &\quad - \frac{e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2)}{e^{\epsilon} - 1} \end{aligned} \tag{21}$$

$$\begin{aligned} P_{-C\leftarrow x}(\mathcal{M}_3) &= P_{C\leftarrow -x}(\mathcal{M}_3) \\ &= P_{-C\leftarrow x}(\mathcal{M}_2) \\ &\quad - \frac{e^{\epsilon} P_{-C\leftarrow 1}(\mathcal{M}_2) - P_{-C\leftarrow -1}(\mathcal{M}_2)}{e^{\epsilon} - 1} \end{aligned} \tag{22}$$

$$\begin{aligned} P_{0\leftarrow x}(\mathcal{M}_3) &= 1 - P_{C\leftarrow x}(\mathcal{M}_3) - P_{-C\leftarrow x}(\mathcal{M}_3) \\ &= P_{0\leftarrow x}(\mathcal{M}_2) \\ &\quad + \frac{2(e^{\epsilon} P_{C\leftarrow -1}(\mathcal{M}_2) - P_{C\leftarrow 1}(\mathcal{M}_2))}{e^{\epsilon} - 1}. \end{aligned} \tag{23}$$

Moreover, the mechanism $\mathcal{M}_3$ has a worst case noise variance smaller than that of $\mathcal{M}_2$, while $\mathcal{M}_3$ still satisfies the requirements in (9a)–(9c).

Proof: The proof details are given in Appendix B of the supplementary material. ∎

We have proved that the symmetric mechanism $\mathcal{M}_3$ has a smaller worst case noise variance than that of mechanism $\mathcal{M}_2$ in Lemma 2, and then we use mechanism $\mathcal{M}_3$ to obtain

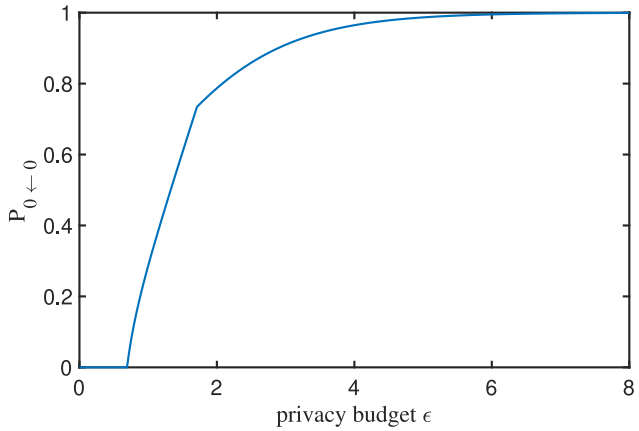


Fig. 3. Optimal $P_{0\leftarrow 0}$ if the privacy budget $\epsilon \in [0, 8]$.

the relation between $P_{0\leftarrow 1}$ and $P_{0\leftarrow 0}$ to find the minimum variance. From Lemma 3 below, we enforce

$$P_{0\leftarrow 0} = e^{\epsilon} P_{0\leftarrow 1}. \tag{24}$$

Then, we use the following Lemma 3 to obtain the relation between $P_{0\leftarrow 1}$ and $P_{0\leftarrow 0}$, so that we can obtain $P_{C\leftarrow x}$, $P_{0\leftarrow x}$, and $P_{-C\leftarrow x}$ using $P_{0\leftarrow 0}$.

Lemma 3: Given $P_{0\leftarrow 0}$, the variance of the output given input x is a strictly decreasing function of $P_{0\leftarrow 1}$ and hence is minimized when $P_{0\leftarrow 1} = P_{0\leftarrow 0}/e^{\epsilon}$.

Proof: The proof details are given in Appendix C of the supplementary material. ∎

Lemma 3 shows that we get the minimum variance when $P_{0\leftarrow 1} = P_{0\leftarrow 0}/e^{\epsilon}$. Hence, we replace $e^{\epsilon} P_{0\leftarrow 1}$ with $P_{0\leftarrow 0}$. Then, the variance is equivalent to

$$\begin{aligned}&\mathrm{Var}[Y|X=x]\\ &= \left(\frac{e^{\epsilon}+1}{(e^{\epsilon}-1)(1-\frac{P_{0\leftarrow 0}}{e^{\epsilon}})}\right)^2 \left(1 - P_{0\leftarrow 0} + (P_{0\leftarrow 0} - \frac{P_{0\leftarrow 0}}{e^{\epsilon}})|x|\right) - x^2.\end{aligned} \tag{25}$$

Complete details for obtaining (25) are in Appendix C of the supplementary material.

Next, we use Lemma 4 to obtain the optimal $P_{0\leftarrow 0}$ in `Three-Outputs` to achieve the minimum worst case variance as follows.

Lemma 4: The optimal $P_{0\leftarrow 0}$ to minimize the $\max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ is defined by (5).

Proof: The proof details are given in Appendix E of the supplementary material. ∎

Remark 1: Fig. 3 displays how $P_{0\leftarrow 0}$ changes with ϵ in (5). When the privacy budget ϵ is small, $P_{0\leftarrow 0} = 0$. Thus, `Three-Outputs` is equivalent to Duchi *et al.*'s [22] solution when $P_{0\leftarrow 0} = 0$. However, as the privacy budget ϵ increases, $P_{0\leftarrow 0}$ increases, which means that the probability of outputting true value increases.

By summarizing above, we obtain $P_{-C\leftarrow x}$, $P_{C\leftarrow x}$ and $P_{0\leftarrow x}$ from (2)–(3) using $P_{0\leftarrow 0}$.

Algorithm 3: `PM-OPT` Mechanism for 1-D Numeric Data Under LDP

Input: tuple $x \in [-1, 1]$ and privacy parameter ϵ.
Output: tuple $Y \in [-A, A]$.

```
Value t is calculated in the (30);
Sample u uniformly at random from [0, 1];
if u < e^ε/(t + e^ε) then
    Sample Y uniformly at random from
    [L(ε, x, t), R(ε, x, t)];
else
    Sample Y uniformly at random from
    [−A, L(ε, x, t)) ∪ (R(ε, x, t), A];
return Y;
```

Then, we can calculate the optimal $P_{0\leftarrow 0}$ to obtain the minimum worst case noise variance of `Three-Outputs` as follows:

Lemma 5: The minimum worst case noise variance of `Three-Outputs` is obtained when $P_{0\leftarrow 0}$ satisfies (5).

Proof: The proof details are given in Appendix F of the supplementary material. ∎

A Clarification About `Three-Outputs` Versus `Four-Outputs`: One may wonder why we consider a perturbation mechanism with three outputs (i.e., our `Three-Outputs`) instead of a perturbation mechanism with four outputs (referred to as `Four-Outputs`), since using two bits to encode the output of a perturbation mechanism can represent four outputs. The reason is as follows. The approach to design `Four-Outputs` is similar to that for `Three-Outputs`, but the detailed analysis for `Four-Outputs` will be even more tedious than that for `Three-Outputs` (which is already quite complex). Given above reasons, we elaborate `Three-Outputs` but not `Four-Outputs` in this article.

B. *PM-OPT Mechanism*

Now, we advocate an optimal `PM` (`PM-OPT`) as shown in Algorithm 3 to get a small worst case variance when the privacy budget is large. As shown in Fig. 1, `Three-Outputs`'s worst case noise variance is smaller than `PM`'s when the privacy budget $\epsilon < 3.2$. But it loses the advantage when the privacy budget $\epsilon \geq 3.2$. As the privacy budget increases, Kairouz *et al.* [12] suggest to send more information using more output possibilities. Besides, we observe that it is possible to improve Wang *et al.*'s [8] `PM` to achieve a smaller worst case noise variance. Thus, inspired by them, we propose an optimal `PM` named as `PM-OPT` with a smaller worst case noise variance than `PM`.

For a true input $x \in [-1, 1]$, the probability density function of the randomized output $Y \in [-A, A]$ after applying LDP is given by

$$\mathbb{F}\big[Y = y|x\big] = \begin{cases} c, & \text{for } y \in [L(\epsilon, x, t), R(\epsilon, x, t)] \quad (26a)\\ d, & \text{for } y \in [-A, L(\epsilon, x, t)) \cup (R(\epsilon, x, t), A] \quad (26b)\end{cases}$$

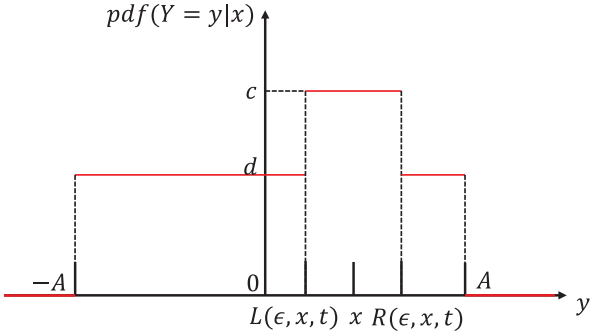


Fig. 4. Probability density function $\mathbb{F}[Y = y|x]$ of the randomized output Y after applying ϵ-LDP.

where

$$c = \frac{e^\epsilon t(e^\epsilon - 1)}{2(t + e^\epsilon)^2} \tag{27}$$

$$d = \frac{t(e^\epsilon - 1)}{2(t + e^\epsilon)^2} \tag{28}$$

$$A = \frac{(e^\epsilon + t)(t + 1)}{t(e^\epsilon - 1)} \tag{29}$$

$$L(\epsilon, x, t) = \frac{(e^\epsilon + t)(xt - 1)}{t(e^\epsilon - 1)}$$

$$R(\epsilon, x, t) = \frac{(e^\epsilon + t)(xt + 1)}{t(e^\epsilon - 1)}$$

and

$$t = \begin{cases} \frac{1}{2}\sqrt{e^{2\epsilon} + 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}}} + \\ \frac{1}{2}\sqrt{2e^{2\epsilon} - 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}} + \frac{4e^\epsilon - 2e^{3\epsilon}}{\sqrt{e^{2\epsilon} + 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}}}}} \\ -\frac{e^\epsilon}{2}, \text{ if } \epsilon < \ln\sqrt{2} & \text{(30a)} \\ -\frac{1}{2}\sqrt{e^{2\epsilon} + 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}}} + \\ \frac{1}{2}\sqrt{2e^{2\epsilon} - 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}} - \frac{4e^\epsilon - 2e^{3\epsilon}}{\sqrt{e^{2\epsilon} + 2^{2/3}\sqrt[3]{e^{2\epsilon} - e^{4\epsilon}}}}} \\ -\frac{e^\epsilon}{2}, \text{ if } \epsilon > \ln\sqrt{2} & \text{(30b)} \\ \frac{\sqrt{3 + 2\sqrt{3}} - 1}{\sqrt{2}}, \text{ if } \epsilon = \ln\sqrt{2}. & \text{(30c)} \end{cases}$$

The meaning of t can be seen from $(t-1)/(t+1) = L(\epsilon, 1, t)/R(\epsilon, 1, t)$. When the input is $x = 1$, the length of the higher probability density function $\mathbb{F}[Y = y|x] = e^\epsilon t(e^\epsilon - 1)/(2(t + e^\epsilon)^2)$ is $R(\epsilon, 1, t) - L(\epsilon, 1, t)$. $R(\epsilon, 1, t)$ is the right boundary, and $L(\epsilon, 1, t)$ is the left boundary. If $0 < t < \infty$, we can derive $\lim_{t\to 0}(t-1)/(t+1) = -1$, meaning the right boundary is opposite to the left boundary if t is close to 0. Since $\lim_{t\to\infty}(t-1)/(t+1) = 1$, it means that the right boundary is equal to the left boundary when t is close to ∞.

Moreover, Fig. 4 illustrates that the probability density function of (26) contains three pieces. If $y \in [L(\epsilon, x, t), R(\epsilon, x, t)]$, the probability density function is equal to c which is higher than other two pieces $y \in [-A, L(\epsilon, x, t))$ and $y \in (R(\epsilon, x, t), A]$. We calculate the probability of a variable Y falling in the interval $[L(\epsilon, x, t), R(\epsilon, x, t)]$ as $\mathbb{P}[L(\epsilon, x, t) \le Y \le R(\epsilon, x, t)] = \int_{L(\epsilon,x,t)}^{R(\epsilon,x,t)} c\, dY = e^\epsilon/(t + e^\epsilon)$.

Furthermore, we use the following lemmas to establish how we get the value t in (26).

Lemma 6: Algorithm 3 achieves ϵ-LDP. Given an input value x, it returns a noisy value Y with $\mathbb{E}[Y|x] = x$ and

$$\mathrm{Var}[Y|x] = \frac{t+1}{e^\epsilon - 1}x^2 + \frac{(t + e^\epsilon)\big((t+1)^3 + e^\epsilon - 1\big)}{3t^2(e^\epsilon - 1)^2}. \tag{31}$$

Proof: The proof details are given in Appendix I of the supplementary material. ∎

Thus, when $x = 1$, we obtain the worst case noise variance as follows:

$$\max_{x\in[-1,1]} \mathrm{Var}[Y|x] = \frac{t+1}{e^\epsilon - 1} + \frac{(t + e^\epsilon)\big((t+1)^3 + e^\epsilon - 1\big)}{3t^2(e^\epsilon - 1)^2}. \tag{32}$$

Then, we obtain the optimal t in Lemma 7 to minimize (32).

Lemma 7: The optimal t for $\min_t \max_{x\in[-1,1]} \mathrm{Var}[Y|x]$ is (30).

Proof: By computing the first-order derivative and second-order derivative of $\min_t \max_{x\in[-1,1]} \mathrm{Var}[Y|x]$, we get the optimal t. The proof details are given in Appendix F of the supplementary material. ∎

Algorithm 4: PM-SUB Mechanism for 1-D Numeric Data Under LDP

Input: tuple $x \in [-1, 1]$ and privacy parameter ϵ.
Output: tuple $Y \in [-A, A]$.

```
Sample u uniformly at random from [0, 1];
if u < e^ε/(e^{ε/3} + e^ε) then
    Sample Y uniformly at random from
    [(e^ε + e^{ε/3})(xe^{ε/3} − 1)/e^{ε/3}/(e^ε − 1),
    (e^ε + e^{ε/3})(xe^{ε/3} + 1)/e^{ε/3}/(e^ε − 1)];
else
    Sample Y uniformly at random from
    [−A, (e^ε + e^{ε/3})(xe^{ε/3} − 1)/e^{ε/3}/(e^ε − 1)) ∪
    ((e^ε + e^{ε/3})(xe^{ε/3} + 1)/e^{ε/3}/(e^ε − 1), A];
return Y;
```

C. PM-SUB Mechanism

We propose a suboptimal PM (PM-SUB) to simplify the sophisticated computation of t in (4) of PM-OPT, and details of PM-SUB are shown in Algorithm 4.

Fig. 1 illustrates that PM-OPT achieves a smaller worst case noise variance compared with PM, but the parameter t for PM-OPT in (30) is complicated to compute. Some vehicles are unable to process the complicated computation. To make t simple for vehicles to implement, we need to find a simple expression for it while ensuring the mechanism's performance. Then, we find that Wang *et al.*'s [8] PM is the case when $t = e^{\epsilon/2}$. Inspired by PM, $\ln t$ and ϵ can be linearly related. Then, we find that $\ln t/\epsilon$ is close to 1/3 [t for PM-OPT in (30)], so we can set $e^{\epsilon/3}$ as t in (26) for a new mechanism named as

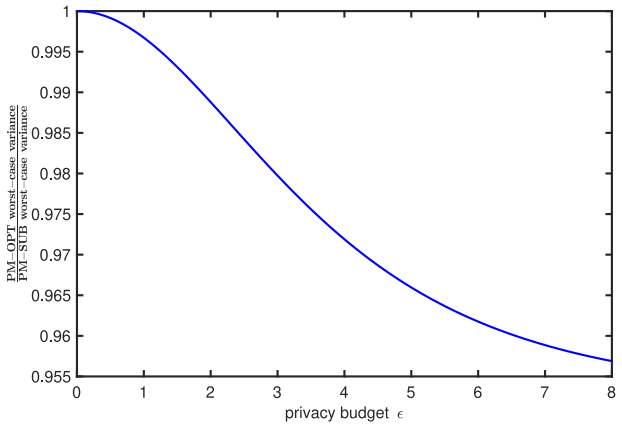


Fig. 5. PM-OPT's worst case noise variance versus PM-SUB's worst case noise variance.

Algorithm 5: Discretization Postprocessing

Input: Perturbed data $y \in [-C, C]$, and domain $[-C, C]$ is separated into $2m$ pieces, where m is a positive integer.

Output: Discrete data Z.

Sample a Bernoulli variable u such that

$$\mathbb{P}[u=1] = \left(\frac{C \cdot (\lfloor \frac{m \cdot y}{C} \rfloor + 1)}{m} - y\right) \cdot \frac{m}{C};$$

if $u = 1$ **then**

  $Z = C \cdot \lfloor \frac{m \cdot y}{C} \rfloor / m;$

else

  $Z = C \cdot (\lfloor \frac{m \cdot y}{C} \rfloor + 1)/m;$

return Z;

PM-SUB. The probability of a variable Y falling in the interval $[L(\epsilon, x, e^{\epsilon/3}), R(\epsilon, x, e^{\epsilon/3})]$ is $e^{\epsilon}/(e^{\epsilon/3} + e^{\epsilon})$, and we give the detail of proof in Appendix J in the supplementary material.

Similar to PM-OPT, we derive the worst case noise variance of PM-SUB from Lemma 6 with $t = e^{\epsilon/3}$ as follows:

$$\max_{x \in [-1,1]} \mathrm{Var}[Y|x] = \frac{5e^{4\epsilon/3}}{3(e^{\epsilon}-1)^2} + \frac{5e^{2\epsilon/3}}{3(e^{\epsilon}-1)^2} + \frac{2e^{\epsilon}}{(e^{\epsilon}-1)^2}. \tag{33}$$

As shown in Fig. 5, PM-SUB's worst case noise variance is close to PM-OPT's, but it is smaller than PM's, which can be observed in Fig. 1.

D. Discretization Postprocessing

Both PM-OPT and PM-SUB's output ranges is $[-1, 1]$ which is continuous, so that there are infinite output possibilities given an input x. Thus, it is difficult to encode their outputs for vehicles. Hence, we consider to apply a postprocessing process to discretize the continuous output range into finite output possibilities. Algorithm 5 shows our discretization postprocessing steps.

The idea of Algorithm 5 is as follows. We discretize the range of output into $2m$ parts due to the symmetric range $[-C, C]$, and then we obtain $2m+1$ output possibilities. After we get a perturbed data y, it will fall into one of $2m$ segments. Then, we categorize it to the left boundary or the right boundary of the segment, which resembles sampling a Bernoulli variable.

Next, we explain how we derive probabilities for the Bernoulli variable. Let the original input be x. A random variable Y represents the intermediate output after the perturbation and a random variable Z represents the output after the discretization. The range of Y is $[-C, C]$. Because the range of output is symmetric with respect to 0, we discretize both $[-C, 0]$ and $[0, C]$ into m parts, where the value of m depends on the user' requirement. Thus, we discretize Y to Z to take only the following $(2m+1)$ values:

$$\left\{ i \times \frac{C}{m} : \text{integer } i \in \{-m, -m+1, \ldots, m\} \right\}. \tag{34}$$

When Y is instantiated as $y \in [-C, C]$, we have the following two cases.

① If y is one of the above $(2m+1)$ values, we set Z as y.

② If y is not one of the above $(2m+1)$ values, and then there exist some integer $k \in \{-m, -m+1, \ldots, m-1\}$ such that $kC/m < y < (k+1)C/m$. In fact, this gives $k < ym/C < k+1$, so we can set $k := \lfloor ym/C \rfloor$. Then conditioning on that Y is instantiated as y, we set Z as kC/m with probability $k+1-ym/C$ and as $(k+1)C/m$ with probability $ym/C - k$, so that the expectation of Z given $Y = y$ equals y [as we will show in (145), this ensures that the expectation of Z given the original input as x equals x].

The following Lemma 8 shows the probability distribution of assigning y with a boundary value in the second case above when the intermediate output y is not one of discrete $(2m+1)$ values.

Lemma 8: After we obtain the intermediate output y after perturbation, we discretize it to a random variable Z equal to kC/m or $(k+1)C/m$ with the following probabilities:

$$\mathbb{P}[Z = z \mid Y = y] = \begin{cases} k+1-\dfrac{ym}{C}, & \text{if } z = \dfrac{kC}{m} \\ \dfrac{ym}{C} - k, & \text{if } z = \dfrac{(k+1)C}{m}. \end{cases} \tag{35}$$

Proof: The proof details are given in Appendix K of the supplementary material. ■

After discretization, the worst case noise variance does not change or get worse proved by Lemma 9 as follows.

Lemma 9: Let LDP mechanism be Mechanism $\mathcal{M}_1$, and discretization algorithm be Mechanism $\mathcal{M}_2$. Let all of output possibilities of Mechanism $\mathcal{M}_1$ be S_1, and output possibilities of Mechanism $\mathcal{M}_2$ be S_2. $S_2 \subset S_1$. When given input x, $\mathcal{M}_1$ and $\mathcal{M}_2$ are unbiased. The worst case noise variance of Mechanism $\mathcal{M}_2$ is greater than or equal to the worst case noise variance of Mechanism $\mathcal{M}_1$.

Proof: The proof details are given in Appendix L of the supplementary material. ■

E. HM-TP Mechanism

Fig. 1 shows that Three-Outputs outperforms PM-SUB when the privacy budget ϵ is small, whereas PM-SUB

achieves a smaller variance if the privacy budget ϵ is large. To fully take advantage of two mechanisms, we combine `Three-Outputs` and `PM-SUB` to create a new hybrid mechanism named as `HM-TP`. Fig. 1 illustrates that `HM-TP` obtains a lower worst case noise variance than other solutions.

Hence, `HM-TP` invokes `PM-SUB` with probability β. Otherwise, it invokes `Three-Outputs`. We define the noisy variance of `HM-TP` as $\mathrm{Var}_{\mathcal{H}}[Y|x]$ given inputs x as follows:

$$\mathrm{Var}_{\mathcal{H}}[Y|x] = \beta \cdot \mathrm{Var}_{\mathcal{P}}[Y|x] + (1-\beta) \cdot \mathrm{Var}_{\mathcal{T}}[Y|x]$$

where $\mathrm{Var}_{\mathcal{P}}[Y|x]$ and $\mathrm{Var}_{\mathcal{T}}[Y|x]$ denote noisy outputs' variances incurred by `PM-SUB` and `Three-Outputs`, respectively. The following lemma presents the value of β.

Lemma 10: The $\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x]$ is minimized when β is (150). Due to the complicated equation of β, we put it in the Appendix in the supplementary material.

Proof: The proof details are given in Appendix M of the supplementary material. ∎

Since we have obtained the probability β, we can calculate the exact expression for the worst case noise variance in Lemma 11 as follows.

Lemma 11: If β satisfies Lemma 10, we obtain the worst case noise variance of `HM-TP` as

$$\max_{x\in[-1,1]} \mathrm{Var}_{\mathcal{H}}[Y|x] = \begin{cases} \mathrm{Var}_{\mathcal{H}}[Y|x^*], \text{ if } 0 < \beta \\ \quad < \dfrac{2(e^\epsilon - a)^2(e^\epsilon - 1) - ae^\epsilon(e^\epsilon+1)^2}{2(e^\epsilon - a)^2(e^\epsilon + t) - ae^\epsilon(e^\epsilon+1)^2}, \\ \max\{\mathrm{Var}_{\mathcal{H}}[Y|0], \mathrm{Var}_{\mathcal{H}}[Y|1]\}, \text{otherwise}, \end{cases}$$

where $x^* := (\beta-1)ae^\epsilon(e^\epsilon+1)^2/(2(e^\epsilon-a)^2(\beta(e^\epsilon+t)-e^\epsilon+1))$ and $a = P_{0\leftarrow 0}$ which is defined in (5).

Proof: The proof details are given in Appendix U of the supplementary material. ∎

VII. Mechanisms for Estimation of Multiple Numeric Attributes

Now, we consider a case in which the user's data record contains $d > 1$ attributes. There are three existing solutions to collect multiple attributes as follows.

1) The straightforward approach which collects each attribute with privacy budget ϵ/d. Based on the composition theorem [47], it satisfies ϵ-LDP after collecting of all attributes. But the added noise can be excessive if d is large [8].
2) Duchi *et al.*'s [22] solution, which is rather complicated, handles numeric attributes only.
3) Wang *et al.*'s [8] solution is the advanced approach that deals with a data tuple containing both numeric and categorical attributes. Their algorithm requires to calculate an optimal $k < d$ based on the single dimensional attribute's ϵ-LDP mechanism, and a user submits selected k dimensional attributes instead of d dimensions.

Thus, we follow Wang *et al.*'s [8] idea to extend Section VI to the case of multidimensional attributes. Algorithm 6 shows the pseudo-code of our extension for our `PM-SUB`, `Three-Outputs`, and `HM-TP`. Given a tuple $x \in [-1,1]^d$, the algorithm returns a perturbed tuple Y that has nonzero value on k attributes, where

Algorithm 6: Mechanism for Multiple-Dimensional Numeric Attributes

Input: tuple $x \in [-1,1]^d$ and privacy parameter ϵ.
Output: tuple $Y \in [-A, A]^d$.

```
Let Y = ⟨0, 0, ..., 0⟩;
Let k = max{1, min{d, ⌊ε/2.5⌋}};
Sample k values uniformly without replacement from {1, 2, ..., d};
for each sampled value j do
    Feed x[t_j] and ε/k as input to PM-SUB, Three-Outputs or HM-TP, and obtain a noisy value y_j;
    Y[t_j] = (d/k) y_j;
return Y;
```

$$k = \max\left\{1, \min\left\{d, \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\right\}\right\} \tag{36}$$

and Appendix S in the supplementary material proves our selected k is optimal after extending `PM-SUB`, `Three-Outputs`, and `HM-TP` to support d dimensional attributes.

Overall, our algorithm for collecting multiple attributes outperforms existing solutions, which is confirmed by our experiments in Section VIII. But `Three-Outputs` uses only one more bit compared with Duchi *et al.*'s [22] solution to encode outputs. Moreover, our `Three-Outputs` obtains a higher accuracy in the high privacy regime (where the privacy budget is small) and saves many bits for encoding since `PM` and `HM`'s continuous output range requires infinite bits to encode, whereas `PM-SUB` and `HM-TP`'s advantages are obvious at a large privacy budget. Furthermore, because vehicles cannot encode continuous range, we discretize the continuous range of outputs to discrete outputs. Our experiments in Section VII-C confirm that we can achieve similar results to algorithms before discretizing by carefully designing the number of discrete parts. Hence, our proposed algorithms are obviously more suitable for vehicles than existing solutions.

Intuitively, Algorithm 6 requires every user to submit k attributes instead of d attributes, such that the privacy budget for each attribute increases from ϵ/d to ϵ/k, which helps to minimize the noisy variance. In addition, by setting k as (36), Algorithm 6 achieves an asymptotically optimal performance while preserving privacy, which we will prove using Lemmas 12 and 13. Lemmas 12 and 13 are proved in the same way as that of Lemmas 4 and 5 in [8].

Lemma 12: Algorithm 6 satisfies ϵ-LDP. In addition, given an input tuple x, it outputs a noisy tuple Y, such that for any $j \in [1, d]$, and each t_j of those k attributes is selected uniformly at random (without replacement) from all d attributes of x, and then $\mathbb{E}[Y[t_j]] = x[t_j]$.

Proof: Algorithm 6 composes k numbers of ϵ-LDP perturbation algorithms; thus, based on composition theorem of

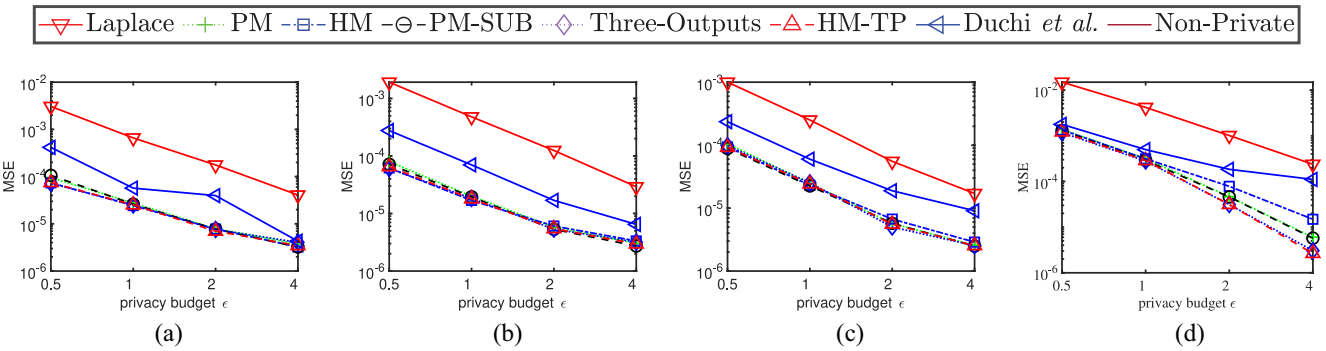


Fig. 6. Result accuracy for mean estimation (on numeric attributes). (a) MX-Numeric. (b) BR-Numeric. (c) WISDM-Numeric. (d) Vehicle-Numeric.

differential mechanism [48], Algorithm 6 satisfies ϵ-LDP. As we can see from Algorithm 6, each perturbed output Y equals to $(d/k)y_j$ with probability k/d or equals to 0 with probability $1 - k/d$. Thus, $\mathbb{E}[Y[t_j]] = k/d \cdot \mathbb{E}[d/k \cdot y_j] = \mathbb{E}[y_j] = x[t_j]$ holds. ■

Lemma 13: For any $j \in [1, d]$, let $Z[t_j] = 1/n \sum_{i=1}^{n} Y[t_j]$ and $X[t_j] = 1/n \sum_{i=1}^{n} x[t_j]$. With at least $1 - \beta$ probability

$$\max_{j \in [1,d]} |Z[t_j] - X[t_j]| = O\left(\frac{\sqrt{d \ln(d/\beta))}}{\epsilon \sqrt{n}}\right).$$

Proof: The proof details are given in Appendix R of the supplementary material. ■

VIII. Experiments

We implemented both existing solutions and our proposed solutions, including `PM-SUB`, `Three-Outputs`, `HM-TP` proposed by us, `PM` and `HM` proposed by Wang *et al.* [8], Duchi *et al.*'s [22] solution and the traditional Laplace mechanism. Our data sets include 1) the WISDM Human Activity Recognition data set [49] is a set of accelerometer data collecting on Android phones from 35 subjects performing six activities, where the domain of the timestamps of the phone's uptime is removed from the data set, and the remaining three numeric attributes are accelerations in x, y, and z directions measured by the Android phone's accelerometer and two categorical attributes; 2) two public data sets extracted from Integrated Public Use Microdata Series [50] contain census records from Brazil (BR) and Mexico (MX). BR includes 4M tuples and 16 attributes, of which six are numerical and ten are categorical. MX contains $4M$ records and 19 attributes, of which five are numerical and 14 are categorical; and 3) a vehicle data set obtained by collecting from a distributed sensor network, including acoustic (microphone), seismic (geophone), and infrared (polarized IR sensor) [51]. The data set contains 98 528 tuples and 101 attributes, where 100 attributes are numerical representing information, such as the raw time series data observed at each sensor and acoustic feature vectors extracted from each sensor's microphone. One attribute is categorical denoting different types of vehicles, which are labeled manually by a human operator to ensure high accuracy. Besides, information about vehicles is gathered to find out the type or brand of the vehicle. The Vehicle data set is also used as the FL benchmark by [52]. We normalize the domain of each numeric attribute to $[-1, 1]$. In our experiments, we report average results over 100 runs.

A. Results on the Mean Values of Numeric Attributes

We estimate the mean of every numeric attribute by collecting a noisy multidimensional tuple from each user. To compare with Wang *et al.*'s [8] mechanisms, we follow their experiments and then divide the total privacy budget ϵ into two parts. Assume a tuple contains d attributes which include d_n numeric attributes and d_c categorical attributes. Then, we allocate $d_n\epsilon/d$ budget to numeric attributes, and $d_c\epsilon/d$ to categorical ones, respectively. Our approach of using LDP for categorical data is same as that of Wang *et al.* [8]. We estimate the mean value for each of the numeric attributes using existing methods: 1) Duchi *et al.*'s [22] solution handles multiple numeric attributes directly; 2) when using the Laplace mechanism, it applies ϵ/d budget to each numeric attribute individually; and 3) `PM` and `HM` are from Wang *et al.* [8]. In Section VII, we evaluate the mean square error (MSE) of the estimated mean values for numeric attributes using our proposed approaches. Fig. 6 presents MSE results as a function of the total budget of ϵ in the data sets (WISDM, MX, BR and Vehicle). To simplify the complexity, we use last 6 numerical attributes of the Vehicle data set to calculate MSE. Overall, our experimental evaluation shows that our proposed approaches outperform existing solutions. `HM-TP` outperforms existing solutions in all settings, whereas `PM-SUB`'s MSE is smaller than `PM`'s when privacy budget ϵ is large, such as 4, and `Three-Outputs`' performance is better at a small privacy budget. Hence, experimental results are in accordance with our theories.

We also run a set of experiments on synthetic data sets that contain numeric attributes only. We create four synthetic data sets, including 16 numeric attributes where each attribute value is obtained by sampling from a Gaussian distribution with mean value $u \in \{0, 1/3, 2/3, 1\}$ and standard deviation of 1/4. By evaluating the MSE in estimating mean values of numeric attributes with our proposed mechanisms, we present our experimental results in Fig. 7. Hereby, we confirm that `PM-SUB`, `Three-Outputs` and `HM-TP` outperform existing solutions.

B. Results on Empirical Risk Minimization

In the following experiments, we evaluate the proposed algorithms' performance using linear regression, logistic

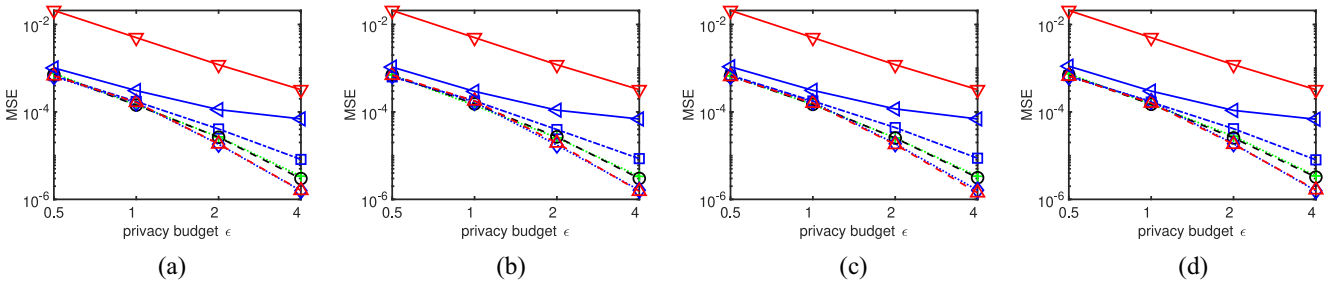


Fig. 7. Result accuracy on synthetic data sets with 16 dimensions, each of which follows a Gaussian distribution $N(\mu, 1/16)$ truncated to $[-1, 1]$. (a) $\mu = 0$. (b) $\mu = 1/3$. (c) $\mu = 2/3$. (d) $\mu = 1$.

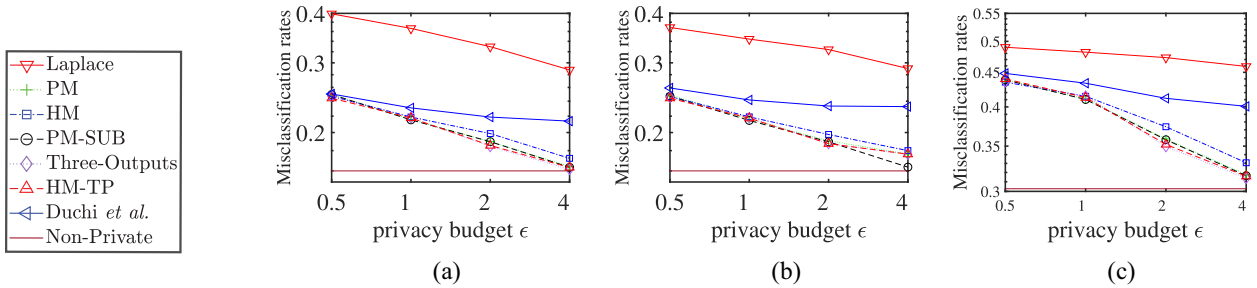


Fig. 8. Logistic regression. (a) MX. (b) BR. (c) WISDN.

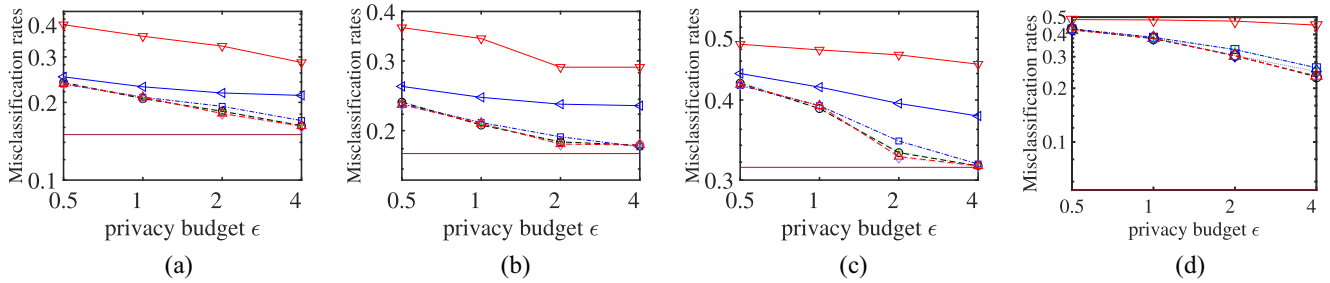


Fig. 9. Support vector machines. (a) MX. (b) BR. (c) WISDN. (d) Vehicle.

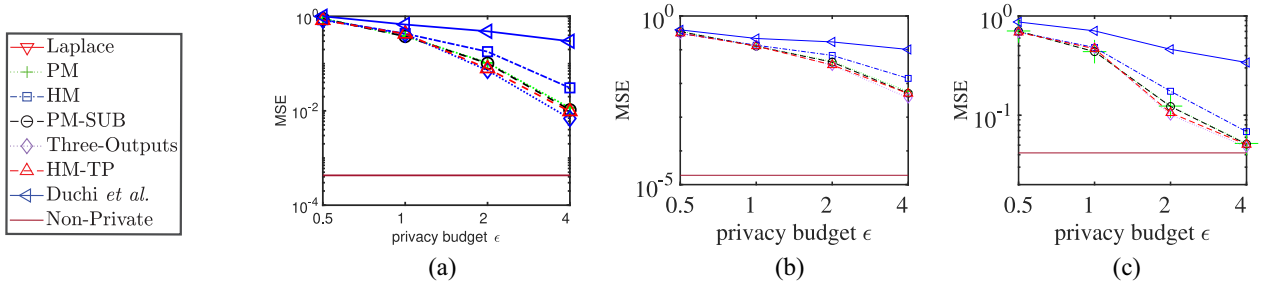


Fig. 10. Linear regression. (a) MX. (b) BR. (c) WISDN.

regression, and SVM classification tasks. We change each categorical attribute t_j with k values into $k - 1$ binary attributes with a domain $\{-1, 1\}$, for example, given t_j 1) 1 represents the lth ($l < k$) value on the lth binary attribute and -1 on each of the rest of $k - 2$ attributes and 2) -1 represents the kth value on all binary attributes. After the transformation, the dimension of WISDN is 43, BR (resp. MX) is 90 (resp. 94) and Vehicle is 101. Since both the BR and MX data sets contain the "total income" attribute, we use it as the dependent variable and consider other attributes as independent variables. The Vehicle data set is used for SVM [51], [52]. Each tuple in the Vehicle data set contains 100-D feature and a binary label.

Consider each tuple of data as the data set of a vehicle, so vehicles calculate gradients and run different LDP mechanisms to generate noisy gradients. Each mini-batch is a group of vehicles. Thus, the centralized aggregator, i.e., cloud server updates the model after each group of vehicles send noisy gradients. The experiment involves 8 competitors: `PM-SUB`, `Three-Outputs`, `HM-TP`, `PM`, `HM`, Duchi *et al.*'s solution, Laplace and a nonprivate setting. We set the regularization factor $\lambda = 10^{-4}$ in all approaches. We use ten-fold

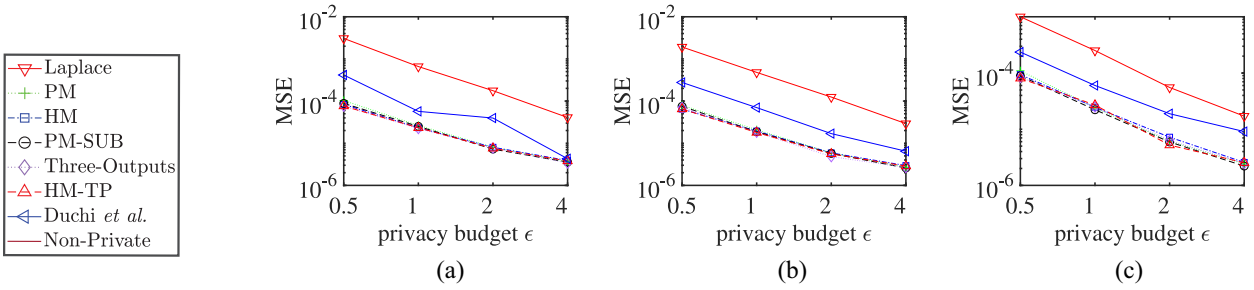


Fig. 11. Result accuracy for mean estimation with discretization post processing on PM, HM, and HM-TP. (a) MX. (b) BR. (c) WISDN.

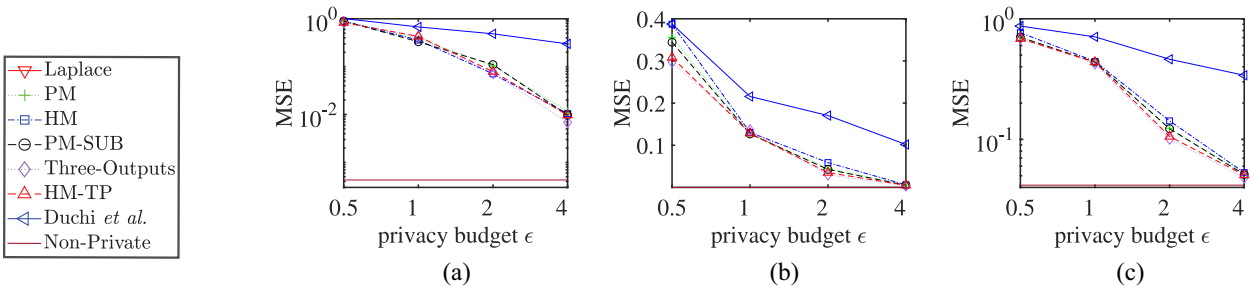


Fig. 12. Linear regression with discretization post processing on PM, HM, and HM-TP (privacy parameter $\epsilon = 4$). (a) MX. (b) BR. (c) WISDN.

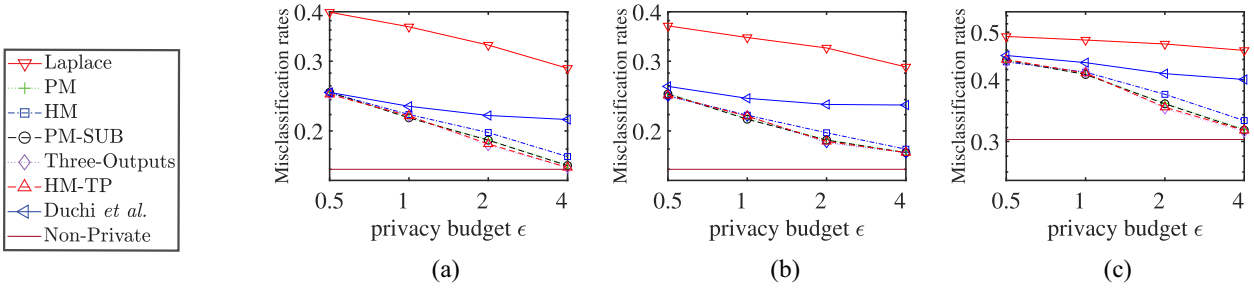


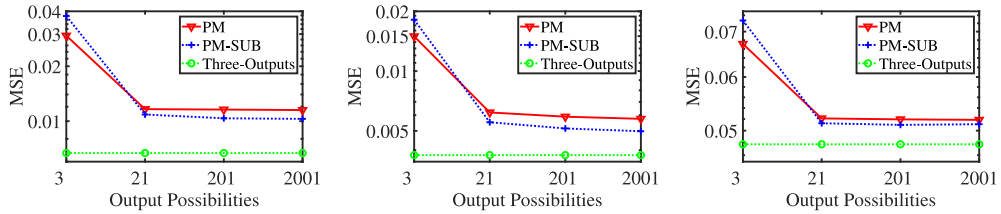
Fig. 13. Logistic regression with discretization post processing on PM, HM, and HM-TP (privacy budget $\epsilon = 4$). (a) MX. (b) BR. (c) WISDN.

cross-validation 5 times to evaluate the performance of each method in each data set. Figs. 8 and 9 show that the proposed mechanisms (PM-SUB, Three-Outputs, and HM-TP) have lower misclassification rates than other mechanisms. Fig. 10 shows the MSE of the linear regression model. We ignore Laplace's result because its MSE clearly exceeds those of other mechanisms. In the selected privacy budgets, our proposed mechanisms (PM-SUB, Three-Outputs, and HM-TP) outperform existing approaches, including Laplace mechanism, Duchi *et al.*'s solution, PM, and HM.

C. Results after Discretization

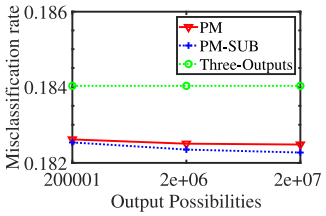
In this section, we add a discretization post processing step in Algorithm 5 to the implementation of mechanisms with continuous range of outputs, including PM, PM-SUB, HM and HM-TP. To confirm that the discretization is effective, we perform the following experiments. We separate the output domain $[-C, C]$ into 2000 segments, and then we have 2001 possible outputs given an initial input x. We add a discretization step to the experiments in Section VIII-A. Fig. 11 displays our experimental results. We confirm that our proposed approaches outperform existing solutions in estimating the mean value using three real-world data sets: 1) WISDM; 2) MX; and 3) BR after discretizing.

In addition, we use log regression and linear regression to evaluate the performance after discretization. We repeat the experiments in Section VIII-B with an additional discretization post processing step. Figs. 12 and 13 present our experimental results. Compared with other approaches, the performance is similar to that before discretizing. Furthermore, Fig. 14 illustrates how the accuracy changes as output possibilities increase. It shows that the misclassification rate of the logistic regression task and the MSE of the linear regression task are related to the size of output possibilities. Although incurring with randomness, we find that the misclassification rate and MSE decrease as the number of output possibilities increases. When there are three output possibilities, it incurs randomness. Moreover, Fig. 15 shows that PM-SUB outperforms Three-Outputs, when the number of output possibilities is large. However, when we discretize the range of outputs into 2000 segments, the performance is satisfactory and similar to the performance with a continuous range of outputs. Hence, our proposed approaches combined with the discretization step help retain the performance while enabling the usage in vehicles.



(a) (b) (c)

Fig. 14. Linear regression with discretization post processing on PM, HM, and HM-TP (privacy budget $\epsilon = 4$). (a) MX. (b) BR. (c) WISDN.



(a) BR

Fig. 15. Support vector machine with discretization post processing on PM, HM, and HM-TP (privacy budget $\epsilon = 5$).

IX. Conclusion

In this article, we propose PM-OPT, PM-SUB, Three-Outputs, and HM-TP LDP mechanisms. These mechanisms effectively preserve the privacy when collecting data records and computing accurate statistics in various data analysis tasks, including estimating the mean frequency and machine learning tasks, such as SVM classification, logistic regression, and linear regression. Moreover, we integrate our proposed LDP mechanisms with FedSGD algorithm to create an LDP-FedSGD algorithm. The LDP-FedSGD algorithm enables the vehicular crowdsourcing applications to train a machine learning model to predict the traffic status while avoiding the privacy threat and reducing the communication cost. More specifically, by leveraging LDP mechanisms, adversaries are unable to deduce the exact location information of vehicles from uploaded gradients. Then, FL enables vehicles to train their local machine learning models using collected data and then send noisy gradients instead of data to the cloud server to obtain a global model. Extensive experiments demonstrate that our proposed approaches are effective and able to perform better than existing solutions. Further, we intend to apply our proposed LDP mechanisms to deep neural network to deal with more complex data analysis tasks.

Acknowledgement

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[2] V. Bindschaedler, R. Shokri, and C. A. Gunter, "Plausible deniability for privacy-preserving data synthesis," *Proc. VLDB Endowm.*, vol. 10, no. 5, pp. 481–492, 2017.

[3] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 603–618.

[4] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, 2019.

[5] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous SGD," 2016. [Online]. Available: arXiv:1604.00981.

[6] J. Duchi, M. J. Wainwright, and M. I. Jordan, "Local privacy and minimax bounds: Sharp rates for probability estimation," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2013, pp. 1529–1537.

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

[8] N. Wang *et al.*, "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. IEEE Int. Conf. Data Eng. (ICDE)*, 2019, pp. 638–649.

[9] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, 2015, pp. 127–135.

[10] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proc. 26th USENIX Security Symp. (USENIX Security)*, 2017, pp. 729–745.

[11] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM Conf. Comput. Commun. Security (CCS)*, 2014, pp. 1054–1067.

[12] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2014, pp. 2879–2887.

[13] L. Ou, Z. Qin, S. Liao, T. Li, and D. Zhang, "Singular spectrum analysis for local differential privacy of classifications in the smart grid," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5246–5255, Jun. 2020.

[14] W. Tang, J. Ren, K. Deng, and Y. Zhang, "Secure data aggregation of lightweight e-healthcare IoT devices with fair incentives," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8714–8726, Oct. 2019.

[15] M. Sun and W. P. Tay, "On the relationship between inference and data privacy in decentralized IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 852–866, 2019.

[16] P. Zhao, G. Zhang, S. Wan, G. Liu, and T. Umer, "A survey of local differential privacy for securing Internet of Vehicles," *J. Supercomput.*, vol. 76, pp. 8391–8412, Dec. 2019.

[17] L. Sun, J. Zhao, and X. Ye, "Distributed clustering in the anonymized space with local differential privacy," 2019. [Online]. Available: arXiv:1906.11441.

[18] M. E. Gursoy, A. Tamersoy, S. Truex, W. Wei, and L. Liu, "Secure and utility-aware data collection with condensed local differential privacy," *IEEE Trans. Depend. Secure Comput.*, early access, Oct. 25, 2019, doi: 10.1109/TDSC.2019.2949041.

[19] S. Ghane, A. Jolfaei, L. Kulik, K. Ramamohanarao, and D. Puthal, "Preserving privacy in the Internet of connected vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 15, 2020, doi: 10.1109/TITS.2020.2964410.

[20] L. Lyu *et al.*, "Towards fair and privacy-preserving federated deep models," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 11, pp. 2524–2541, Nov. 2020.

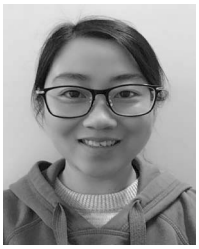
[21] L. Sun, X. Ye, J. Zhao, C. Lu, and M. Yang, "Bisample: Bidirectional sampling for handling missing data with local differential privacy," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2020, pp. 88–104.
[22] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *J. Amer. Stat. Assoc.*, vol. 113, no. 521, pp. 182–201, 2018.
[23] C. Xu, J. Ren, L. She, Y. Zhang, Z. Qin, and K. Ren, "EdgeSanitizer: Locally differentially private deep inference at the edge for mobile data analytics," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5140–5151, Jun. 2019.
[24] W.-S. Choi, M. Tomei, J. R. S. Vicarte, P. K. Hanumolu, and R. Kumar, "Guaranteeing local differential privacy on ultra-low-power systems," in *Proc. ACM/IEEE 45th Annu. Int. Symp. Comput. Archit. (ISCA)*, Los Angeles, CA, USA, 2018, pp. 561–574.
[25] X. He, J. Liu, R. Jin, and H. Dai, "Privacy-aware offloading in mobile-edge computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Singapore, 2017, pp. 1–6.
[26] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4755–4763, Jun. 2019.
[27] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "Local differential privacy for deep learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5827–5842, Jul. 2020.
[28] V. Pihur, "The podium mechanism: Improving on the laplace and staircase mechanisms," 2019. [Online]. Available: arXiv:1905.00191.
[29] L. Zhao *et al.*, "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," *IEEE Trans. Depend. Secure Comput.*, early access, Apr. 14, 2020, doi: 10.1109/TDSC.2020.2986205
[30] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey," 2019. [Online]. Available: arXiv:1909.11875.
[31] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Trans. Ind. Informat.*, vol. 16, no. 10, pp. 6532–6542, Oct. 2020.
[32] S. Lu, Y. Yao, and W. Shi, "Collaborative learning on the edges: A case study on connected vehicles," in *Proc. 2nd USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, 2019, pp. 1–8.
[33] R. Fantacci and B. Picano, "Federated learning framework for mobile edge computing networks," *CAAI Trans. Intell. Technol.*, vol. 5, no. 1, pp. 15–21, 2020.
[34] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," 2019. [Online]. Available: arXiv:1909.00907.
[35] M. Brendan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–14.
[36] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, and Y. Zhou, "A hybrid approach to privacy-preserving federated learning," in *Proc. 12th ACM Workshop Artif. Intell. Security*, 2019, pp. 1–11.
[37] R. Hu, Y. Guo, H. Li, Q. Pei, and Y. Gong, "Personalized federated learning with differential privacy," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9530–9539, Oct. 2020.
[38] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," vol. 32, in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2019, pp. 15479–15488. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/fc0de4e0396fff257ea362983c2dda5a-Paper.pdf
[39] A. Triastcyn and B. Faltings, "Federated learning with Bayesian differential privacy," 2019. [Online]. Available: arXiv:1911.10071.
[40] Y. Wang, Y. Tong, and D. Shi, "Federated latent Dirichlet allocation: A local differential privacy based framework," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 6283–6290.
[41] L. Lyu, J. C. Bezdek, X. He, and J. Jin, "Fog-embedded deep learning for the Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4206–4215, Jul. 2019.
[42] T. Li, Z. Liu, V. Sekar, and V. Smith, "Privacy for free: Communication-efficient learning with differential privacy using sketches," 2019. [Online]. Available: arXiv:1911.00972.
[43] Y. Zhao *et al.*, "Privacy-preserving blockchain-based federated learning for IoT devices," 2019. [Online]. Available: arXiv:1906.10893.
[44] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2020.
[45] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020. [Online]. Available: arXiv:2003.02133.
[46] T. Wang, J. Zhao, H. Yu, J. Liu, X. Yang, X. Ren, and S. Shi, "Privacy-preserving crowd-guided AI decision-making in ethical dilemmas," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, 2019, pp. 1311–1320.
[47] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
[48] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. IEEE Symp. Found. Comput. Sci. (FOCS)*, Providence, RI, USA, 2007, pp. 94–103.
[49] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newslett.*, vol. 12, no. 2, pp. 74–82, 2011.
[50] S. Ruggles *et al.*, *IPUMS USA: Version 10.0*, Minneapolis, MN, USA: IPUMS, 2020. [Online]. Available: https://doi.org/10.18128/D010.V10.0
[51] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.
[52] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019. [Online]. Available: arXiv:1905.10497.

Yang Zhao (Graduate Student Member, IEEE) received the master's degree in electrical engineering from the National University of Singapore, Singapore, in 2015. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Her research interests include federated learning, blockchain, differential privacy, and 6G.

Jun Zhao (Member, IEEE) received the bachelor's degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University (CMU) Pittsburgh, PA, USA (advisors: Virgil Gligor, Osman Yagan; collaborator: Adrian Perrig), affiliating with CMU's renowned CyLab Security and Privacy Institute in 2015.

He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Before joining NTU first as a Postdoctoral Fellow with Xiaokui Xiao and then as a Faculty Member, he was a Postdoctoral Fellow with Arizona State University, Tempe, AZ, USA, as an Arizona Computing Postdoctoral Best Practices Fellow (advisors: Junshan Zhang, Vincent Poor). His research interests include communications, networks, security, and AI.



Mengmeng Yang (Member, IEEE) received the B.Eng. degree from Qingdao Agriculture University, Qingdao, China, in 2011, the M.Sc. degree from Shenyang Normal University, Shenyang, China, in 2014, and the Ph.D. degree in computer science from Deakin University, Geelong, VIC, Australia, in 2019.

She is currently a Research Fellow with the Strategic Centre for Research in Privacy-Preserving Technologies and Systems, Nanyang Technological University, Singapore. Her research interests include privacy preserving, data mining, and machine learning.

Teng Wang (Member, IEEE) received the B.S. degree from the School of Software, Xidian University, Xi'an, China, in 2015, and the Ph.D. degree from the School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, in 2020.

She is currently a Lecturer with the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an. She was a visiting Ph.D. student with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, from 2018 to 2019. Her research interests include mobile crowdsensing systems, privacy-preserving data collection and analysis, and privacy-preserving in machine learning and artificial intelligence.

Ning Wang (Member, IEEE) received the Ph.D. degree in computer software and theory from Northeastern University, Shenyang, China, in 2017.

She is currently a Lecturer with Ocean University of China, Qingdao, China. Her current research interest lies in data privacy protection and big data analytics.

Lingjuan Lyu (Member, IEEE) received the B.E. degree from the Hefei University of Technology, Hefei, China, in 2011, the M.E. degree from the Chinese Academy of Science, Beijing, China, in 2014, and the Ph.D. degree from the University of Melbourne, Melbourne, VIC, Australia, in 2018.

She is currently a Research Fellow with the National University of Singapore, Singapore. She was a Research Fellow (level B3) with Australian National University, Canberra, ACT, Australia. Her current research interests include federated learning, trustworthy AI, edge intelligence, and fairness.

Dr. Lyu was a winner of IBM Ph.D. Fellowship, from 2017 to 2018.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests are in the area of energy harvesting for wireless communication, Internet of Things, and sensor networks.

Kwok-Yan Lam (Senior Member, IEEE) received the B.Sc. (First Class Hons.) degree in computer science from the University of London, London, U.K., in 1987, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1990.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. He was a Professor with the Tsinghua University, Beijing, China, from 2002 to 2010. He has been a Faculty Member of the National University of Singapore, Singapore, and the University of London since 1990. He was a visiting scientist with the Isaac Newton Institute, Cambridge University and a Visiting Professor with the European Institute for Systems Security. His research interests include distributed systems, IoT security infrastructure, distributed protocols for blockchain, biometric cryptography, homeland security, and cybersecurity.

Prof. Lam received the Singapore Foundation Award from the Japanese Chamber of Commerce and Industry in recognition of his Research and Development achievement in Information Security in Singapore in 1998.