Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 18(4): 3006–3033.
DOI: 10.3934/mbe.2021151
Received: 15 September 2020
Accepted: 02 March 2021
Published: 29 March 2021

Research article

Insuring against the perils in distributed learning: privacy-preserving empirical risk minimization

Kwabena Owusu-Agyemang, Zhen Qin, Appiah Benjamin, Hu Xiong and Zhiguang Qin*

University of Electronic Science and Technology of China, School of Information and Software Engineering, China

* **Correspondence:** Email: qinzhen@uestc.edu.cn.

Abstract: Multiple organizations would benefit from collaborative learning models trained over aggregated datasets from various human activity recognition applications without privacy leakages. Two of the prevailing privacy-preserving protocols, secure multi-party computation and differential privacy, however, are still confronted with serious privacy leakages: lack of provision for privacy guarantee about individual data and insufficient protection against inference attacks on the resultant models. To mitigate the aforementioned shortfalls, we propose privacy-preserving architecture to explore the potential of secure multi-party computation and differential privacy. We utilize the inherent prospects of output perturbation and gradient perturbation in our differential privacy method, and progress with an innovation for both techniques in the distributed learning domain. Data owners collaboratively aggregate the locally trained models inside a secure multi-party computation domain in the output perturbation algorithm, and later inject appreciable statistical noise before exposing the classifier. We inject noise during every iterative update to collaboratively train a global model in our gradient perturbation algorithm. The utility guarantee of our gradient perturbation method is determined by an expected curvature relative to the minimum curvature. With the application of expected curvature, we theoretically justify the advantage of gradient perturbation in our proposed algorithm, therefore closing existing gap between practice and theory. Validation of our algorithm on real-world human recognition activity datasets establishes that our protocol incurs minimal computational overhead, provides substantial utility gains for typical security and privacy guarantees.

Keywords: Internet of things; differential privacy; fully homomorphic encryption; privacy-preserving; secure multi-party computations; human activity recognition

1. Introduction

Lately, Distributed Machine Learning (DML) [1] architectures have gradually produce remarkable performance across a wide variety of domains in Industrial Internet of Things (IoT), including facial recognition, machine translation, object detection, and object classification. As the size of datasets grow, learning from private industrial internet of things datasets has frequently been confronted with challenging privacy risks in numerous data analytic applications. The learning algorithms are expected to effectively learn from the data whiles providing a certain level of privacy-preserving guarantee for the users' confidential data. Meanwhile adversaries might infer information from the training dataset used in the classifier whiles the parameters in the classifier are also capable of revealing certain sensitive information in the dataset. In situations involving the training of DNN classifiers, parameters of the model can also store the private information on the training dataset. These kinds of attacks have demonstrated to be practical both in the federated and centralized domain, therefore posing a serious threat to diverse privacy-preserving settings.

Consequently, Differential Privacy (DP) [2] is a robust privacy concept for statistical data privacy with the potential to provide meaningful guarantees irrespective of what an adversary learns beforehand about the individuals' dataset in the centralized domain where an individual company owns all the data. DP delivers unarguable privacy guarantee to ensure the impact of any single data record becomes extremely insignificant. In many real-world architectures, differential privacy has been deployed and embraced by commercial enterprises and the U.S. Census Bureau. Numerous machine learning algorithms have been combined with the modification of differential privacy to satisfy its privacy-preserving requirements, prominent among the widely used supervised learning models is Empirical Risk Minimization (ERM). It's Differentially Private version is Differential Private Empirical Risk Minimization [3,4] (DP-ERM) which can be defined as follows:

Definition 1. *(DP-ERM). Given a dataset $D = \{z_1, z_2 \cdots, z_n\}$ from a data universe $\mathcal{X}$ and a closed convex set $C \subseteq \mathbb{R}^p$, DP-ERM is to find $x^{priv} \in C$ so as to minimize the empirical risk, i.e.,*

$$x_* \in \arg\min_{x \in C} F^r(x, D) = F(x, D) + r(x) = \frac{1}{n} \sum_{i=1}^{n} f(x, z_i) + r(x)$$

with the guarantee of being differentially private, where f is the loss function and r is some simple (non)smooth convex function called regularizer. When the inputs are drawn i.i.d from an specified underlying distribution $\mathcal{P}$ on $\mathcal{X}$, we as well consider population risk $\mathbb{E}_{z\sim\mathcal{P}}[f(x, z)]$. If the loss function is convex, the utility of the algorithm is measured by the expected excess empirical risk, i.e.,

$$\mathbb{E}_{\mathcal{A}}\left[F^r\left(x^{\text{priv}}, D\right)\right] - \min_{x \in C} F^r(x, D),$$

or the expected excess population risk (generalization error), i.e.,

$$\mathbb{E}_{z\sim\mathcal{P},\mathcal{A}}\left[f\left(x^{\text{priv}}, z\right)\right] - \min_{x \in C} \mathbb{E}_{z\sim\mathcal{P}}[f(x, z)],$$

where the expectation of $\mathcal{A}$ is taking over all the randomness of the algorithm.

Empirical risk minimization (ERM) plays a crucial role amid all machine learning models and it covers a diversity of machine learning tasks. As one demonstrate the ability to perform ERM privately, with the application of differentially private algorithms for a wide range of machine learning problems, e.g., regression and classification becomes a straightforward deal. It is appropriate to incorporate randomness into the method to ensure privacy within this machine learning model. Basically, based on the time of introducing noise, there are three methods to introduce randomness: output perturbation [5] (OP), gradient perturbation [6] (GP) and objective perturbation (ObjP).

Output perturbation as a variant of Laplace mechanism initially executes the learning algorithm that is similar to the non-private setting, afterward injects noise to the output parameter. Objective perturbation perturbs the objective function of the ERM i.e., empirical loss and later activates our perturbed objective minimizer whiles providing precise solutions to the current problem where the stability of the accurate solutions plays a very critical role in the analytical process. Gradient perturbation interferes with every interim change. The composition theorem of differential privacy guarantees the entire learning process to become distinctly differentially private as the individual update becomes differentially private. Knifer et al. [7] dominated the extension of objective perturbation to enable them prove similar output for more general cases, predominantly for high-dimensional learning.

Aside ERM-DP, secure multi-party has also been one of the preferred privacy-preserving machine learning methods. Protocols for secure learning in multi-party setting enable individual data owners to collaboratively execute a statistical function together over their confidential inputs with the application of cryptographic primitives such as Fully homomorphic encryption, oblivious transfer, and secrete sharing. During this procedure, each of the individual parties can acquire the correct results and none of the data owners can get any knowledge than the data inferred from the public results. In the last few years, secure multi-party computation has been widely used to achieve privacy-preserving distributed data mining since it is more efficient and effective than other approaches. Existing secure multi-party approaches has explored different threat models and it's applications. In real-life scenarios, secure multi-party computation is capable of supporting floating-point and fix-point operations, whiles controlling the implementation of linear complexities of arithmetic computations. Due to its potential, the study of secure multi-party computation to preserve the privacy of distributed data mining has attracted a great deal of interest in recent years. Currently the focus has been directed towards practical achievement of an efficient distributed machine learning with the application of secure multi-party computation primitives, in some domains, these approaches have demonstrated to scale to learning tasks with numerous records.

Despite the individual success of previous works in ERM-DP and Secure MPC in terms of utility, there are still much work to do from a practical perspective when the fundamental problem of secure multi-party computation meets differential privacy in the distributed domain in the following scenarios:

a) The challenge becomes more critical in circumstances where data is owned by diverse organizations with the aim of collaboratively learning from their sensitive data. Consider in epidemiology research where researchers and doctors in collaboration with the hospitals use the epidemiological information generated from a large number of patient cases to make an accurate diagnosis, treatment, plan and evaluate strategies for disease prevention and also serving as a guide for the management of patients with infected diseases. It is imperative to demonstrate independent scientific training of models on these private data to aid in the identification, risk assessment, evaluate interventions to reduce risk disease infection whiles preserving the privacy of patient data.

b) Unlike approaches using the individual algorithms from secure multi-party computations or differential privacy, these algorithms basically protect the training dataset throughout the learning procedure without providing any protection against membership inference attacks on the resultant classifier (i.e., models).
c) In the distributed setting, since privacy noise affects the optimization procedure, it is important to establish the latest and more tighter utility guarantee for our gradient perturbation schemes to overcome the mismatch existing between empirical observation of gradient perturbation and its theoretical guarantee as compared with the other existing objective and OP approach which plays a critical role in ERM-DP meeting MPC.

DNN has proven to be remarkably effective to numerous machine learning tasks; with a defined parameterized function from inputs to outputs which is a composition of multiple layers of fundamental building blocks. These blocks include simple nonlinear function and affine transformations. With the variation of the parameters of these building blocks, DNN models can be trained from such a parameterized function with aim the of fitting any given finite set of input to predict an output from the samples. More specifically, definition of a loss function ℓ denotes the penalty for mismatching the training dataset is required. The $\ell(\vartheta)$ on parameters ϑ is the average of the loss over the training dataset. Training of the model consist of finding ϑ to produce an appreciable loss, optimistically the minimum loss which is extremely hard to anticipate to attain accurate global minimum in practice. In a complex network, the loss function ℓ is usually non-convex and difficult to minimize. Practically the minimization is basically done by the mini-batch stochastic gradient descent algorithm. In this algorithm we explore the potential of gradient descent in our proposed algorithm. In our secure multi-party domain with the integration of zero-concentrated differential privacy to achieve a privacy-preserving of our datasets and model, we investigate the effectiveness of the application of output perturbation and gradient perturbation in our proposed scheme.

In our threat model, we consider honest-but-curious (semi-honest) data providers who wish to collaboratively train a model without exposing their individual inputs to other data owners. Data providers in this threat model do not collude to temper with the combined functionality or inject garbage input data, they are capable of passively inferring about inputs of other data providers depending on the implementation of the algorithm. We apply [8] and [9] to securely aggregate local classifiers and their gradients in the cloud service provider. Secure multi-party computation algorithms involve two or multiple data owners to collaboratively execute a function of their confidential inputs, without exposing any details about the data inputs other than their data size and whatever can be inferred from the exposed intermediate results. Existing secure multi-party approaches are capable of securely computing functions on aggregated data has greatly been influenced all these years by Yao's garbled circuits protocol. Advancement in secure multi-party computation has efficiently improved its implementation making it more practical to implement two-party protocols possessing millions of higher dimensional data inputs [10]; and also applicable in the global scale involving multi-party protocols with malicious level security for smaller data inputs [11]. Ma et al. [8] demonstrated that secure aggregation of DNN local classifiers using secured multi-party computation protocols is practical. In their paper they used a two-party computation involving non-colluding servers with a semi-honest threat model. Existing approaches has successfully used similar methods to scale multi-party regression [4]. In this work we can use this method [8] to achieve our secured multi-party model aggregation in the cloud. For circumstances where there is high risk of collusion, numerous secure multi-party protocols can be used to an individual honest data provider even if all other data owners are malicious. The focus of this paper is not to improve or evaluate the execution of the secure multi-party computation but rather combine MPC primitives and machine learning in our proposed algorithm for secure learning in the multi-party domain.

In this paper, we present a combination of secure multi-party computation and differentially-private distributed machine learning primitives with the application of both gradient perturbation and output perturbation where the injection of statistical noise is inside the secure multi-party computation. In our proposed algorithm, our output perturbation method aggregates the locally trained classifiers whiles achieving DP with the injection of Laplace statistical noise to the aggregated model parameters. The focus of our proposed algorithm is dominated by gradient perturbation since it comes with numerous potentials over objective perturbation and output perturbation. Furthermore, GP does not demand strong assumptions on the objective function because it simply needs to bound the sensitivity of gradient update rather than the entire learning process. Additionally, gradient perturbation can discharge the noisy gradient during every iteration devoid of destructing the privacy guarantee as DP [12] is invulnerable to post-processing. Therefore, making gradient perturbation the preferred option for applications such as optimization of privacy-preserving machine learning in the distributed domain [4]. Eventually, gradient perturbation frequently attains an improved empirical utility than objective perturbation or output perturbations for DP-ERM.

In our proposed gradient perturbation algorithm, data owners collaborate to execute an iterative, gradient-based learning method with the aim of securely aggregating the local gradients from the trained model during each iteration. Moreover, our gradient perturbation method may not experience severe accuracy degradation instead desired an individual implementation of secure multi-party primitive per iteration. Therefore, making our proposed protocol an accuracy closer to their non-privately trained existing approaches where no statistical noise is injected and the confidentiality of the dataset is not provided.

Contributions : We therefore summarized our contributions in three folds:

- We propose distributed privacy-preserving gradient descent algorithm, which is a combination of two stages output perturbation and Gradient Perturbation algorithm to solve differentially private ERM with massive privately-owned datasets. In our framework, we privately train accurate machine learning classifiers in the distributed domain where the noise is injected inside a secure multi-party computation.
- For our gradient perturbation in our architecture, we introduce an expected curvature which has the potential of characterizing the optimization property with precision as the noise is injected into the model at each gradient update in the iterative process. We establish the utility guarantee for this framework which is grounded on the expected curvature instead of the normal minimum curvature.
- The performance of our algorithm is evaluated with the application of real-world human activity recognition datasets. With the implementation of regularized linear regression and logistic regression models for our regression and classification. The results establishes that MS-DPGD produces models that are very close to non-private models with reference to the accuracy of the model and its generalization error in the distributed machine learning protocols.

The rest of this paper is organized as follows. Section 2 introduces the Related works. In Section

3, we give details of the Preliminaries about differential privacy, Zero-Concentrated Differential Privacy and secure multi-party computation that are exploited in our proposed algorithm. Then Section 4 proposes a novel distributed privacy-preserving protocol of a statistical model. Moreover, we give the performance evaluation results for the application of our proposed algorithm to linear regression computational overhead, accuracy, security trade-offs and scalability in Section 5. We therefore conclude this paper in Section 6.

2. Related works

Distributed Machine Learning (DML) [1] as a decentralized machine learning theory enables distributed training on a large scale of a dataset in edge devices including electronic meters, internet of things environment and sensors smartphones where no individual node is capable of getting the intelligent decision from a massive dataset within an appreciable period of time. DML technique has earned a remarkable reputation in numerous pragmatic areas such as visual object detection, health records, big data analysis, control policies, and medical text extraction. Regrettably, as the number of distributed data owners increases, the guarantee for the security of the datasets from the individual data owners becomes extremely difficult. This lack of security will increase the threat that adversaries attack the dataset preceded by the manipulation of the intermediate training result. Therefore, affecting data integrity which is a key component in training machine learning models. Adversarial data attacks [13] is one of the distinctive ways with the aim of corrupting the machine learning models by contaminating data during the training phase. Consider in a scenario where newly generated datasets are expected to be updated periodically by the data owners for improving the models, the adversary is likely to gain more chances of poisoning the dataset, posing a severe threat in the distributed machine learning models. This kind of threat is considered one of the most imperative emerging security threats against machine learning models. Since the adversarial attack has the potential of misleading the diverse machine learning methods, a widely applicable DML protection mechanism is urgently required to be investigated.

There has been numerous works conducted on privacy-preserving DML, most of the research approaches were greatly stimulated by privacy-preserving machine learning and data mining. The existing literature on privacy-preserving DML basically falls into two major categories: cryptography-based technique and perturbation-based methods.

The cryptography-base methods typically incorporate cryptographic tools to preserve the privacy of the datasets. Secure multi-party computation [14] may potentially address these security threat in the DML system. Though acquiring information with the aggregation of the dataset from multi-parties is a critical task for machine learning, in a real-world business setting, the prevention of privacy leakages while carrying out this privacy-preserving task is a very crucial requirement for secure learning in multi-party setting. Private learning in multi-party domain [15] problem refers to collaborative execution of a statistical function together by multiple data owners. After the computation, individual data owners acquire accurate results and no one can get more knowledge than the dataset inferred from the public intermediate results. Secure multi-party algorithms are basically cryptographic-based methods that apply a typical cryptographic technique to perform these distributed machine learning tasks. The data owners try not to expose any knowledge on original data except that can be inferred from the output [16] of the distributed machine learning task. Over the past few years, secure multi-party computation has widely been applied to achieve privacy-preserving in distributed machine since it is more effective and efficient than other privacy-preserving algorithms. Secure MPC is capable of supporting floating-point and fix-point arithmetic operations [17], this arithmetic functions can be executed with controlled linear complexity [18]. Owing to these benefits, exploring the potential of secure multi-party computation required for privacy-preserving in distributed machine learning has gained considerable attention over the past few years. Initial proposals to secure two-party (2PC) computation was first introduced by Yao [19] in 1982. Subsequently, Goldreich et al. [20] generalized and stretched 2PC method to the secure multi-party computer (MPC) problem. Secure MPC later gain so much research attention finding practical ways of exploring the potential in this domain. Gentry [21] with the aid of homomorphic encryption with ideal lattices primitive was the first to introduce secure MPC. This was later preceded by numerous researchers proposing various secure MPC implementations, including Lost-cost multi-party Computation for Dishonest Majority [22] Semi-homomorphic Encryption [23] and Active-Secure Two-Party Computation [24]. To the highest degree, these algorithms can be categorized into two major methods such as secret sharing and homomorphic encryption. Gentry et al. [21] achieve the initial scheme with multiplicative and additive homomorphism respectively, this will require long period of time to execute the complex circuits during the performance of the inevitable elimination of the noise. To the contrary, secrete sharing primitive [23] is capable of calculating infinite times of any multiplication and addition with additional exchange of datasets. In a typical example of a two-party domain, Bansal et al. [25] applied secure scalar and secret sharing to protect privacy leakages during the training process which is not trivial to be extended to the secure multi-party models. Yuan et al. [26] propose a privacy-preserving back-propagation algorithm for secure multi-party deep learning over arbitrarily partitioned datasets grounded on BGN homomorphic encryption [27]. Nevertheless, the primitive requires all the data owners to be online and interactively collaborate to decrypt the ciphertext of the intervening parameters during each iteration. In [28], Hesamifard et al. propose privacy-preserving machine learning algorithm using encrypted data to make encrypted predictions in the cloud. Since fully homomorphic encryption(FHE) [21] generate high computational complexities, they proposed a confidentially binary classification-based method to find a trade-off between the degree of the polynomial approximation and the secure performance of the training model. Li et al. [29] with the aid of multi-key fully homomorphic encryption (MK-FHE) [30] also propose a privacy-preserving multi-party deep learning primitive, in this setting, the individual data owners encrypts their datasets with different public keys and outsource them to the cloud server, the cloud server therefore train the deep learning classifiers with the application of the MK-FHE. Based on the relevant literature above, it is obvious that most of the cryptography-based algorithms use fully homomorphic or multi-key fully homomorphic encryption primitives to encrypt the entire dataset before outsourcing it to the cloud server.

With the addition of noise to the raw dataset, perturbation-based method is able to protect the privacy of the dataset. Agrawal et al. [31] proposes an algorithm that injects elaborately designed noise to the training data while preserving the statistical properties to enable the training of Naive Bayes classifier. With the gradual explosion of digitized dataset, Fong et al. [32] offered a privacy-preserving learning algorithm by transforming the original dataset into groups of unreal datasets whiles preserving the accuracy for the learning model. Furthermore, this algorithm ensures that the original data samples cannot be reconstructed without the whole group of constructed datasets. DP is a strong golden standard to guarantee privacy-preserving for algorithms on aggregated datasets, which is widely applied in privacy-preserving DML to ensure that the dominance of any single data owners record is insignificant. DP has been utilized in many existing applications by large-scale businesses such as US Census Bureau. A typical distributed machine learning strategy applied in this domain is Empirical Risk Minimization (ERM), where the average error of the trained model over the aggregated datasets is minimized. There have been numerous proposals [33] to advance the works on privacy-preserving algorithms for ERM problems with the application of variations of differential privacy. This paradigm is known as Differentially Private Empirical Risk Minimization (DP-ERM) [3]. DP-ERM reduces the empirical risk whiles providing the assurance the intermediate results of the learning model is differentially private based on the aggregated training dataset. This privacy-preserving guarantee ensures strong protection against potential adversaries such as inference attacks. To guarantee privacy-preserving in this domain, it is always essential to launch randomness to the training protocol. Based on the time for noise injection, there are mainly three ways to initiate randomness: objective perturbation, output perturbation, and gradient perturbation. This work introduces differentially-private DML algorithm with the application of both gradient perturbation and output perturbation whiles injecting the noise within the secure multi-party computation.

3. Preliminaries and background

The sensitive representative datasets from different data owners required for the collaborating training of the deep learning models are the fundamental component of the privacy-preserving machine learning classifiers in our secure multi-party domain. It is very important to prevent this sensitive individual information in the datasets from privacy leakages during the training process of the machine learning classifier. Interestingly, the adversary is capable of creating model inversion attacks by inferring features of the training datasets which may lead to privacy disclosure. Consequently, the integration of privacy-preserving strategies into machine learning models in the secure multi-party settings is a feasible strategy for alleviating the vulnerability to privacy. This section introduces necessary background for our analyses, which include empirical risk minimization (ERM), differential privacy (which involves the zero-concentrated differential with their notations) and basic assumptions in secure multi-scheme computation as applied in our proposed architecture. The main objective of our proposed architecture presents an iterative differential privacy-preserving DML algorithm from the D to enable us prevent leakage of sensitive information in the training dataset, which accept $D = \{d_1, d_2, ..., d_n\}$ as input to accurately outputs y_i as the predicted target. Table 1 provide summary of the notations applied throughout this work.

3.1. Differential privacy

Differential privacy is also integrated into machine learning and deep learning algorithms as a promising technique for privacy preservation to maintain the privacy of training data and models. It delivers a solid privacy assurance to ensure that adversaries cannot infer from the inclusion or exclusion of a record in the database irrespective of their possession of the information about all records except the target one. Details of differential privacy is shown as follows:

Table 1. Notations in our proposed architecture.

Notations	Description
D	Database of n records
$(\mathbf{x}_i, y_i)$	The i-th record in database D
ϑ	The parameter vector of neural networks
ϑ^*	The optimal model parameter ϑ
$\mathcal{L}(\vartheta)$	The loss function on database D
ϵ	The privacy budget of neural networks
R_j	Average relevance
Δ	Sensitivity
$\mathrm{Lap}(\cdot)$	Laplace distribution
$g(\mathbf{x}_i)$	Gradients
$\tilde{g}(\mathbf{x}_i)$	The noisy gradients
α	Relevance ratio

Definition 2. *(ϵ, δ)-Differential Privacy [2]. A randomized mechanism $\mathcal{M}$ satisfy (ϵ, δ)-DP if for every event $\mathcal{S} \subseteq$ range $(\mathcal{M})$ and for all $D, D' \in D^n$,*

$$(Pr[\mathcal{M}(D) \in S] \leq \exp(e)Pr[\mathcal{M}(D') \in S] + \delta). \tag{3.1}$$

When $\delta = 0$, $\mathcal{M}$ accomplishes pure differential privacy by providing stronger privacy protection than approximate DP with $\Delta > 0$. We can add noise sampled form Gaussian and Laplace distributions respectively to achieve ϵ-DP and (ϵ, δ)-DP where the statistical noise is proportional to ℓ_2 norm sensitivity of $\mathcal{M}$; given as $\Delta\mathcal{M} = ||\mathcal{M}(D) - \mathcal{M}(D')||$.

Definition 3. *(ℓ_1 and ℓ_2 norm sensitivity). Let $q : D^n \leftarrow \mathbb{R}^d$ be a query function. The ℓ_1(resp. ℓ_2) sensitivity of q, denoted by $\Delta_1(q)$ (resp., $\Delta_2(q)$) is defined as follows:*

$$\Delta_1(q) = \max_{D \sim D} ||q(D) - q(D')||_1, \; \Delta_2(q) = \max_{D \sim D} ||q(D) - q(D'||_2 \tag{3.2}$$

The ℓ_1 and ℓ_2 sensitivities constitutes the maximum change in the output value of q (over all possible neighbouring databases in D^n) when an individual's dataset is altered.

Theorem 1. *Let $\epsilon \in (0, 1)$ be arbitrary and q be a query function with ℓ_2 sensitivity of $\Delta_2(q)$. The Gaussian Mechanism, which returns $q(D) + N(0, o^2)$ with*

$$\sigma \geq \frac{\Delta_2(q)}{\epsilon} \sqrt{2 \ln(1.25/\delta)} \tag{3.3}$$

is (ϵ, δ)-DP. A critical property of DP is its privacy guarantee reduces gracefully under the composition. The most basic composition result shows that the privacy loss grows linearly under k-fold composition [12]. This implies, if we sequentially apply an (ϵ, δ)-differential privacy algorithm n times on the same data, the resulting process is $(n\epsilon, n\delta)$. Dwork et al. [2] provided a booting method to construct an improved privacy-preserving synopsis on the queries with an advance composition; the loss function grows sub-linearly at the rate of $\sqrt{n}$.

Theorem 2. *For all* $\epsilon, \delta, \delta' \geq 0$ *the class of* (ϵ, δ)*-differential private mechanisms satisfy*$(\epsilon', n\delta + \delta')$*-differential privacy under k-fold adaptive composition for*

$$\epsilon' = \sqrt{2n} In(\frac{1}{\delta'})\epsilon + n\epsilon(e^{\epsilon - 1}) \tag{3.4}$$

as stated in the advance composition [2]

3.2. Zero-concentrated differential privacy

Whiles differential privacy is suitable for algorithms such as output perturbation, it is not the preferred option for gradient perturbation that require repeated sampling of statistical noise during the iterative training approach. Bun and Stainke [34] provided zero-concentrated DP (zCDP) which has a tight composition bound and superior to gradient perturbation. We define ρ -zCDP by the introduction of the privacy loss random variable as applied in the definition of zCDP.

Definition 4. *For an output* $0 \in$ *range*$(\mathcal{M})$*, the privacy loss random variable* $\mathcal{Z}$ *of the mechanism* $\mathcal{M}$ *is defined as follows:*

$$Z = \log \frac{\mathbb{P}[\mathcal{M}(D) = \circ]}{\mathbb{P}\left[\mathcal{M}(D') = \circ\right]} \tag{3.5}$$

ρ-zCDP therefore imposes a bound on the moment generating function of the privacy loss $\mathcal{Z}$ and requires it to be tightly concentrated around zero mean, hence it is unlikely to distinguish $\mathcal{D}$ from $\mathcal{D}\prime$.Formally, it important to satisfy the following:

$$e^{D_\alpha(M(D)\|\mathcal{M}(D'))} = \mathbb{E}\left[e^{(\alpha-1)Z}\right] \leq e^{(\alpha-1)\alpha\rho}, \forall\alpha \in (1, \infty) \tag{3.6}$$

where $D_\alpha\left(M(D)\|\mathcal{M}(D')\right)$ is the α-Rényi divergence. In this work, we apply the resulting zCDP composition.

Lemma 1. *[34] If two mechanisms satisfy* ρ_1 *-zCDP and* ρ_2 *-zCDP, then their composition satisfy* $(\rho_1 + \rho_1)$*-zCDP.*

If two mechanisms satisfy ρ_1 -zCDP and ρ_2 -zCDP, then their composition satisfy $(\rho_1 + \rho_2)$-zCDP

Lemma 2. *[34] The Gaussian mechanism returns* $q(\mathcal{D}) + \mathcal{N}(0, \sigma^2)$ *satisfies* $\Delta_2(q)^2/(2\sigma^2)$*-zCDP*

Lemma 3. *[34] If* $\mathcal{M}$ *satisfies* ϵ*-DP, then* $\mathcal{M}$ *satisfy* $(\frac{1}{2}\epsilon^2)$*-zCDP, then* $\mathcal{M}$ *is* $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$*-DP for any* $\delta > 0$.

4. MS-DPGD architecture

4.1. Model aggregation with output perturbation

In our proposed algorithm, we extend the differential privacy bound of [35] to the secure multi-party domain, where adequate noise is injected into the model to preserve the privacy of individual data owners and final output throughout the multi-party training process. Our model aggregation with the output the perturbation model is represented in Figure 1.

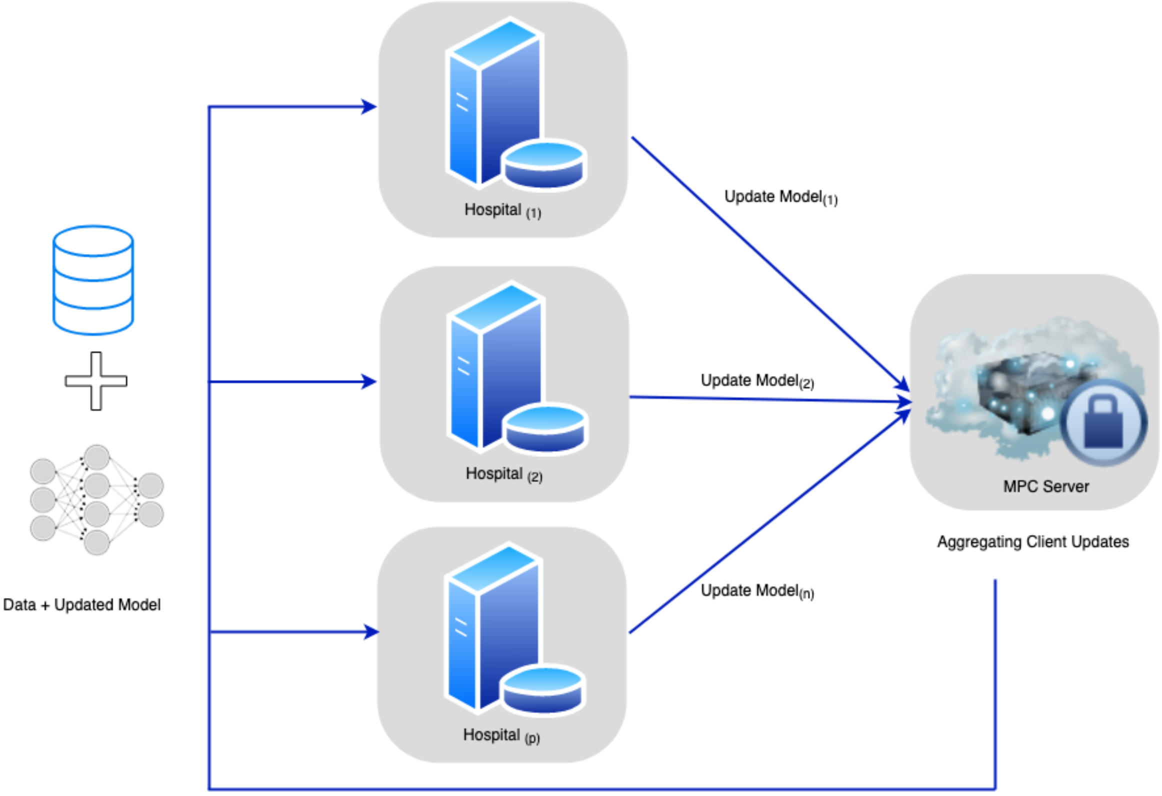

Figure 1. Workflow of our output perturbation method.

Theorem 3. *Given a set of n_j of size for training data instances each having a dataset D_j from k parties, with the data instances lying in a ball of radius, the sensitivity of regularized classifier is at most $\frac{2}{\lambda n}$, If ϑ_1 and ϑ_2 are classifiers trained on adjacent datasets $D, D^{'}$ of size n_j with regularization parameter λ.*

$$\|\boldsymbol{\vartheta}_1 - \boldsymbol{\vartheta}_2\|_1 \leq \frac{2}{\lambda n}$$

We therefore obtain their corresponding local model estimator $\hat{\vartheta}^j$ with a given k data owners each possessing a dataset D_j of size n_j:

$$J_D(\vartheta) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\vartheta, (x_i, y_i)\right) + \lambda N(\vartheta),$$

we obtain $\widehat{\vartheta}^{(j)}$ as its corresponding model estimator.

$$\widehat{\vartheta}_{(privacy)} = \frac{1}{k}\sum_{j=1}^{k} \widehat{\vartheta}^{(j)} + \eta$$

is the perturbed aggregate model estimator; where η is the Laplace noise injected into the aggregated model estimator to achieve differential privacy. We therefore adopt Ma et al.'s [8] secure model aggregation for our secure MPC model in the cloud server. The theory below provides a bound on the magnitude of noise required to achieve DP.

Theorem 4. *If*

$$\widehat{\vartheta}_{(privacy)} = \frac{1}{k}\sum_{j=1}^{k}\widehat{\vartheta}^{(j)} + \eta$$

is the perturbed aggregate model estimator whiles

$$\widehat{\vartheta}^{(j)} = \arg\min_{\vartheta}\frac{1}{n_j}\sum_{i=1}^{n_j}\ell\left(\vartheta, x_i^{(j)}, y_i^{(j)}\right) + \lambda N(\vartheta)$$

with the data lying in a unit ball and $\ell(\cdot)$ is G-Lipschitz , then $\widehat{\vartheta}_{(privacy)}$ is ϵ-differentially private if :

$$\eta = \mathrm{Lap}\left(\frac{2G}{kn_{(1)}\lambda\epsilon}\right)$$

$n_{(1)}$represent the size of the smallest data set among the k parties, λ is the regularization parameter and ϵ is the differential privacy budget.

Proof. Let there be k parties such that one record of party j changes in the neighboring data sets then:

$$\frac{\mathbf{Pr}(\widehat{\vartheta}|D)}{\mathbf{Pr}(\widehat{\vartheta}|D')} = \frac{\mathbf{Pr}\left(\frac{1}{k}\sum_{i\neq j}\widehat{\vartheta}^{(i)} + \frac{1}{k}\widehat{\vartheta}^{(j)} + \eta|D\right)}{\mathbf{Pr}\left(\frac{1}{k}\sum_{i\neq j}\widehat{\vartheta}^{(i)} + \frac{1}{k}\widehat{\vartheta}'^{(j)} + \eta|D'\right)} \quad = \frac{\exp\left[\frac{k\cdot n_{(1)}\epsilon\lambda}{2G}\frac{\left\|\widehat{\vartheta}^{(j)}\right\|}{k}\right]}{\exp\left[\frac{k\cdot n_{(1)}\epsilon\lambda}{2G}\frac{\left\|\widehat{\vartheta}^{(j)}\right\|}{k}\right]} \leq \exp\left[\frac{n_{(1)}\epsilon\lambda}{2G}\left\|\widehat{\vartheta}^{(j)} - \widehat{\vartheta}'^{(j)}\right\|\right]$$

$$\leq \exp\left[\frac{n_{(1)}\epsilon\lambda}{2G}\frac{2G}{n_j\lambda}\right] \leq \exp(\epsilon)$$

Provision of a bound on the excess empirical risk and true risk is similar to [36] and [4] with our bounds tighter than both of them as we require very fewer differential privacy noise.

Theorem 5. *If a perturbed aggregated model estimator*

$$\widehat{\vartheta}_{privacy} = \frac{1}{k}\sum_{j=1}^{k}\widehat{\vartheta}^{(j)} + \eta, \ \textit{where}\ \widehat{\vartheta}^{(j)} = \arg\min_{\vartheta}\frac{1}{n_j}\sum_{i=1}^{n_j}\ell\left(\vartheta, x_i^{(j)}, y_i^{(j)}\right) + \lambda N(\vartheta)$$

and an optimal model estimator ϑ^* trained on the centralized data such that the data lie in a unit ball and $\ell(\cdot)$ is G-Lipschitz and L-smooth, then the bound on excess empirical risk is given as:

$$J\left(\widehat{\vartheta}_{privacy}\right) \leq J(\vartheta^*) + A_1\frac{G^2(\lambda + L)}{n_{(1)}^2\lambda^2}(m^2 + \frac{d^2\log^2(d/\delta)}{m_1^2\epsilon^2} + \frac{d\log(d/\delta)}{\epsilon})$$

where A_1 is an absolute constant. We proof Theorem 5 by following Pathak et al. [36]. We therefore make a provision bound on the Laplace random vector given in Lemma 5 as proven in [37]. To achieve a tighter bound our choice of sensitivity bound is given as $2G/(kn_{(1)}\lambda)$.The full proof is given in the second Proof of Theorem 6. To enable us prove Theorem 5, a provision of a bound on the Laplace random vector as stated in the Lemma below and as proven in [37].

Lemma 4. *Given a d-dimensional random variable $\eta \sim \text{Lap}(\beta)$ with $P(\eta) = \frac{1}{2\beta}e^{-\frac{\|\eta\|_1}{\beta}}$ with probability $1-\delta$ the ℓ_2-norm of the random variable is bounded as $\|\eta\| \leq d\beta \log\left(\frac{d}{\delta}\right)$ For any differentiable and convex objective function, [37] propose the following Lemma to bound the sensitivity of our model estimator:*

Lemma 5. *Let $G(\vartheta)$ and $g(\vartheta)$ be two differentiable convex functions of ϑ. If $\vartheta_1 = \arg\min_\vartheta G(\vartheta)$ and $\vartheta_2 = \arg\min_\vartheta G(\vartheta) + g(\vartheta)$, then $\|\vartheta_1 - \vartheta_2\| \leq \frac{g_1}{G_2}$ where $g_1 = \max_\vartheta \|\nabla g(\vartheta)\|$ and $G_2 = \min_v \min_\vartheta v^T \nabla^2 G(\vartheta) v$ for any unit vector $v \in \mathbb{R}^d$.*

Theorem 6 is expressed to bound the excess risk involving the optimal model estimator and non-private model estimator in the centralized domain. We therefore use this Theorem to prove Theorem 5.

Theorem 6. *With an aggregate model estimator,*

$$\widehat{\vartheta} = \frac{1}{k}\sum_{j=1}^{m} \widehat{\vartheta}^{(j)} \text{ where } \widehat{\vartheta}^{(j)} = \arg\min_{\theta} \frac{1}{n_j}\sum_{i=1}^{n_j} \ell(\theta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\vartheta)$$

and an optimal model estimator ϑ^* trained on the centralized data such that the data lie in a unit ball and $\ell(\cdot)$ is G-Lipschitz, we obtain:

$$\|\widehat{\vartheta} - \vartheta^*\| \leq \frac{G(k-1)}{n_{(1)}\lambda}.$$

Proof. For data provider P_j, the local model estimator is stated as:

$$\widehat{\vartheta}^{(j)} = \arg\min_{\vartheta} \frac{1}{n_j}\sum_{i=1}^{n_j} \ell(\vartheta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\vartheta) = \arg\min_{\vartheta} G(\vartheta)$$

Therefore the centralized model estimator is however stated as:

$$\vartheta^* = \arg\min_{\vartheta} \frac{1}{n_j}\sum_{i=1}^{n_j} \ell(\vartheta, x_i^{(j)}, y_i^{(j)}) + \sum_{l\neq j}\frac{1}{n_l}\sum_{i=1}^{n_l} \ell(\vartheta, x_i^{(l)}, y_i^{(l)}) + \lambda N(\vartheta) = \arg\min_{\vartheta} G(\vartheta) + g(\vartheta)$$

Thus we obtain the following values of G_2 and g_1:

$$G_2 = \min_v \min_\vartheta \|v^\top \nabla^2 G(\vartheta) v\| = \min_v \min_\vartheta \|v^\top (\frac{1}{n_j}\|\nabla^2 \ell(\vartheta, x_i^{(j)}, y_i^{(j)})\| + \lambda.1)v\| \geq \lambda$$

$$g_1 = \max_\vartheta \|\nabla g(\vartheta)\| = \max_\vartheta \sum_{l\neq j}\frac{1}{n_l}\|\nabla \ell(\vartheta, x_i^{(l)}, y_i^{(l)})\| \leq G\sum_{l\neq j}\frac{1}{n_l}$$

With the application of Lemma 5. We obtain $\|\widehat{\vartheta}^{(j)} - \vartheta^*\| = \frac{G}{\lambda}\sum_{l\neq j}\frac{1}{n_l}$. By using the triangular inequality, we however obtain:

$$\left\|\widehat{\vartheta} - \vartheta^*\right\| \leq \frac{1}{k}\sum_j \|\widehat{\vartheta}^{(j)} - \vartheta^*\| = \frac{G}{k\lambda}\sum_j\sum_{l\neq j}\frac{1}{n_l} = \frac{G(k-1)}{k\lambda}\sum_j \frac{1}{n_j} \leq \frac{G(m-1)}{n_{(1)}\lambda}$$

Proof. We also continue by using Taylor's Expansion to prove Theorem 5 with the application of Lemma 4 and Theorem 6. We therefore obtain:

$$J(\widehat{\vartheta}_{privacy}) = J(\vartheta^*) + (\widehat{\vartheta}_{privacy} - \vartheta^*)\nabla J(\vartheta^*) + \frac{1}{2}(\widehat{\vartheta}_{privacy} - \vartheta^*)\nabla^2 J(\vartheta)(\widehat{\vartheta}_{privacy} - \vartheta^*)$$

with $\theta = \alpha\widehat{\vartheta}_{privacy} + (1-\alpha)\vartheta^*$for some$\alpha \in [0,1]$. By definition $\nabla J(\vartheta^*) = 0$, we therefore obtain:

$$J(\widehat{\vartheta}_{privacy}) - J(\vartheta^*) \le \frac{1}{2}\|\widehat{\vartheta}_{privacy} - \vartheta^*\|^2 \cdot \|\nabla^2 J(\vartheta)\|$$

since $\ell(\cdot)$ is L -smooth, thus we get $\|\nabla^2 J(\vartheta)\| \le \lambda + L$, therefore

$$\begin{aligned} J(\widehat{\vartheta}_{privacy}) &\le J(\vartheta^*) + \frac{\lambda+L}{2}\|\hat{\vartheta} - \vartheta^* + \eta\|^2 \le J(\vartheta^*) + \frac{\lambda+L}{2}[\|\widehat{\vartheta} - \vartheta^*\|^2 + \|\eta\|^2 + 2(\widehat{\vartheta} - \vartheta^*)^\top \eta] \\ &\le J(\vartheta^*) + \frac{\lambda+L}{2}\left[\left\|\widehat{\vartheta} - \vartheta^*\right\|^2 + \|\eta\|^2 + 2\left\|\widehat{\vartheta} - \vartheta^*\right\| \cdot \|\eta\|\right] \end{aligned}$$

By Theorem 6 and Lemma 4, We therefore get

$$\begin{aligned} J(\widehat{\vartheta}_{privacy}) &\le J(\vartheta^*) + \frac{G^2(k-1)^2(\lambda+L)}{2n_{(1)}^2\lambda^2} + \frac{2G^2d^2(\lambda+L)}{k^2n_{(1)}^2\lambda^2\epsilon^2}\log^2(\frac{d}{\delta}) + \frac{2G^2d(k-1)(\lambda+L)}{kn_{(1)}^2\epsilon\lambda^2}\log(\frac{d}{\delta}) \\ &\le J(\vartheta^*) + A_1\frac{G^2(\lambda+L)}{n_{(1)}^2\lambda^2}(k^2 + \frac{d^2\log^2(d/\delta)}{k^2\epsilon^2} + \frac{d\log(d/\delta)}{\epsilon}) \end{aligned}$$

with A_1 is an absolute constant

4.2. Iterative gradient perturbation with expected curvature

With the gradient perturbation in our proposed Multi-scheme Distributed Privacy-preserving Gradient Descent algorithm, we give a considerable attention to a centralized private empirical risk minimization to adopt per-iteration privacy budget for k entities, individually with a dataset $\mathcal{D}_j$ of volume n_j of independent observations.

$$J_D(\vartheta) = \min_{\vartheta} \frac{1}{m}\sum_{j=1}^{m}\frac{1}{n_j}\sum_{i=1}^{n_j}\ell(\vartheta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\vartheta) \tag{4.1}$$

Data owners can collaboratively train a differentially private classifier by adopting the per-iteration privacy budget by the injection of noise into the aggregated gradient inside the secure MPC settings to make individual iteration progress towards an optimal solution. It is important to note that the regularization term in Eq (4.1) possesses no privacy guarantee and does not have any privacy implications since it is independent of the datasets. Our iterative gradient perturbation method is represented in Figure 2.

Theorem 7. *Given ϑ_T as the centralized classifier estimator which is derived from minimizing $J_D(\vartheta)$ later T iterations of gradient descent method executed collaboratively by k data owners with datasets D^j of size n_j with each data instance $(x_i^j, y_i^j) \in D^j$ reside in a unit ball and $\ell(\vartheta)$ is G-Lipschitz and*

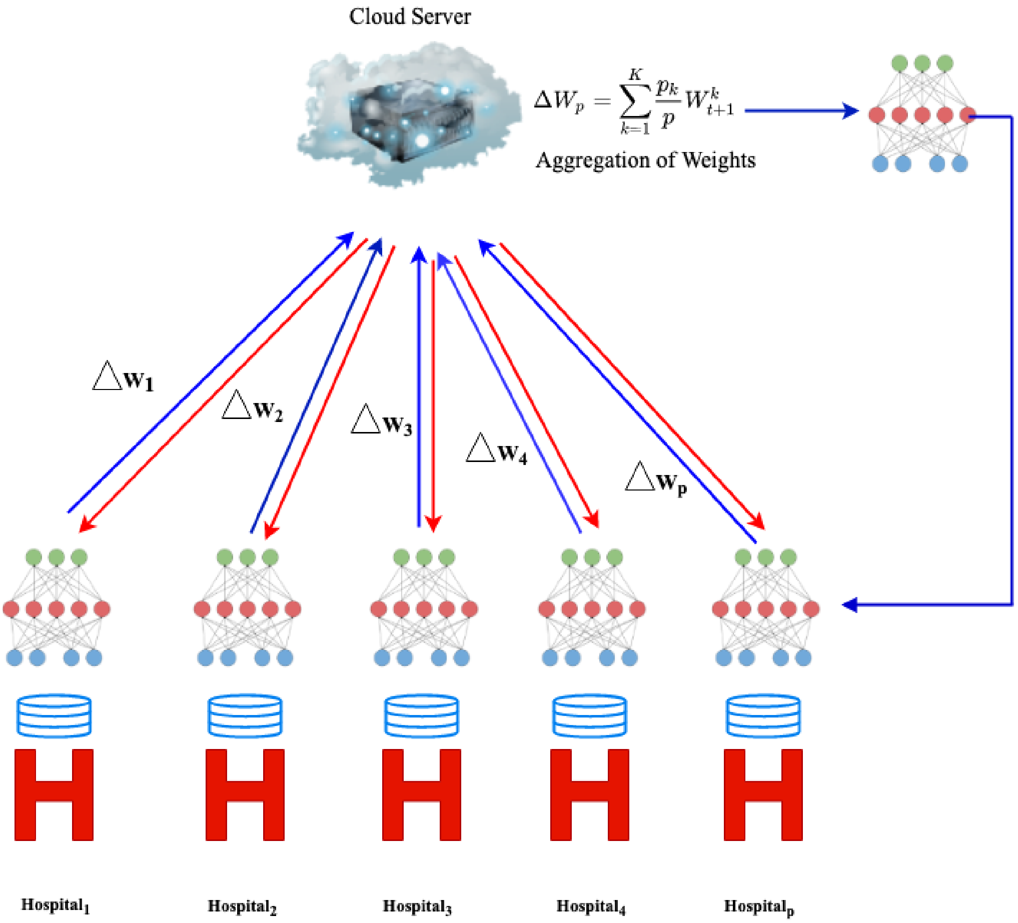

Figure 2. Gradient perturbation workflow.

L-smooth over $\vartheta \in C$. In this setting our learning rate is $\frac{1}{L}$ whiles the gradients are also perturbed with statistical noise $z \in \mathcal{N}(0, \sigma^2 I_d)$, we can therefore conclude that ϑ_T is (ϵ, δ)-differentially private if:

$$\sigma^2 = \frac{8G^2 T \log(1/\delta)}{\epsilon^2 k^2 n_1^2} \tag{4.2}$$

the smallest size of n in our k data owners is represented as n_1.

Proof. At a gradient of step t;

$$G_t = \nabla J(\vartheta, D) + \mathcal{N}(0, \sigma^2 I_p) = \frac{1}{k}\sum_{j=1}^{k} \frac{1}{n_j}\sum_{i=1}^{n_j} \nabla \ell(\vartheta, x_i^j, y_i^j) + \mathcal{N}(0, \sigma^2 I_p)$$

In our assumption only a data instance of one party is capable of changing in the neighbouring D and D' datasets. Our sensitivity bound therefore becomes

$$\|\nabla J(\vartheta, D) - \nabla J(\vartheta, D')\| \le \frac{2G}{mn_{(1)}}$$

Hence,with the application of Lemma 1, V_t is ρ-zCDP with $\rho = \frac{2G^2}{k^2 n_1^2 \sigma^2}$. Based on Lemma 2, we noticed that ϑ_{Tot} is $T\rho$-zCDP. Using Lemma 3 we therefore obtain $\epsilon = T\rho + 2\sqrt{T\rho\log(1/\delta)}$. To solve roots of this equation as follows:

$$\rho \approx \frac{\epsilon^2}{4T\log(1/\delta)} \Longrightarrow \sigma^2 = \frac{8G^2 T\log(1/\delta)}{\epsilon^2 k^2 n_1^2}$$

Hence, ϑ_{Tot} is (ϵ, δ) -differentially private for the above value of σ^2. Furthermore, we discovered that for each intermediate model estimator ϑ_t differential privacy is also ensured. ϑ_t as our intermediate estimator at every iteration $t \in [1, T]$ is $(\sqrt{t/T}\epsilon, \delta)$ differentially private as proven in Proof 5.

Proof. With the composition property of Lemma 2, individual ϑ_t is $t\rho$-zCDP. With the application of Lemma 3, ϵ_t as the privacy budget for iteration t is given as $\epsilon_t = t\rho + 2\sqrt{t\rho\log(1/\delta)}$ with $\epsilon = T\rho + 2\sqrt{T\rho\log(1/\delta)}$ as the Total privacy budget is as follows:

$$\Rightarrow \frac{t\epsilon}{T} = t\rho + 2\sqrt{t\rho\log(1/\delta)}(\sqrt{\frac{t}{T}}) = t\rho(1 - \sqrt{\frac{t}{T}}) + \sqrt{\frac{t}{T}}\epsilon_t \Longrightarrow \epsilon_t = \sqrt{\frac{t}{T}}\epsilon + \sqrt{Tt}(\sqrt{\frac{t}{T}} - 1)\rho$$

During the Proof of Theorem 4, we demonstrated that

$$\rho \approx \frac{\epsilon^2}{4T\log(1/\delta)}$$

substitution of ρ, we obtain the relation between ϵ and ϵ_t:

$$\epsilon_t = \sqrt{\frac{t}{T}}\epsilon + \sqrt{\frac{t}{T}}(\sqrt{\frac{t}{T}} - 1)\frac{\epsilon^2}{4\log(1/\delta)} \leq \sqrt{\frac{t}{T}}\epsilon$$

Therefore, ϑ_t as the individual intermediate model estimator is $(\sqrt{t/T}\epsilon, \delta)$-differentially private. In this situation the adversarial attacker is unable to obtain any additional information from the intermediate computations. We therefore provide a theoretical bounds on the true excess risk and excess empirical risk of the proposed algorithm.

Theorem 8. *In the centralized model estimator ϑ_T which is obtained by minimizing $J_D(\vartheta)$ after gradient descent method with T iterations collaboratively implemented by k parties with individual dataset D^j of size n_j with individual data instance $(x_i^j, y_i^j) \in D^j$ residing in a unit ball and $\ell(\vartheta)$ is G-Lipschitz and L-smooth over $\vartheta \in \mathcal{A}$. We bridge the gap amid strongly convex objectives and utility guarantee of non-strongly convex in this setting, by applying expected curvature v to demonstrate that part of the non-strongly convex objectives are capable of achieving the same magnitude of utility guarantee as the strongly convex objectives which will match our empirical observation. We therefore define the expected curvature (Definition 5) and further explain it dependence on only the average curvature.*

Definition 5. *Given a convex function $F : \mathbb{R}^p \rightarrow \mathbb{R}$, has expected curvature v relative to noise $\mathcal{N}\left(0, \sigma^2 \boldsymbol{I}_p\right)$ if for any $\boldsymbol{\vartheta} \in \mathbb{R}^p$ and $\tilde{\boldsymbol{\vartheta}} = \boldsymbol{\vartheta} - z$ with $z \sim \mathcal{N}\left(0, \sigma^2 \boldsymbol{I}_p\right)$, it supports that:*

$$\mathbb{E}\left[\langle \nabla J(\vartheta), \vartheta - \vartheta_* \rangle\right] \geq v\mathbb{E}\left[\|\vartheta - \vartheta_*\|^2\right] \tag{4.3}$$

with the expectation taken based on z

Proof. If J is μ-strongly convex function, we therefore obtain $\nu \geq \mu$ which can be established as $\nu = \mu$ since it always dominates due to the strongly convex definition. The average curvature is represented by ν which is wider than μ. We apply ϑ' to represent the transpose of $\boldsymbol{\vartheta}$.

Let $\boldsymbol{H_\vartheta} = \nabla^2 J(\boldsymbol{\vartheta})$ be the Hessian matrix evaluated at $\boldsymbol{\vartheta}$. We therefore apply Taylor's expansion to Eq (4.3) to approximate its left hand side as follows:

$$\begin{aligned}&\mathbb{E}\left[\left\langle \nabla J(\tilde{\boldsymbol{\vartheta}}), \tilde{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}_*\right\rangle\right] \approx \mathbb{E}\left[\langle \nabla J(\boldsymbol{\vartheta}) - \boldsymbol{H_\vartheta} z, \boldsymbol{\vartheta} - z - \boldsymbol{\vartheta}_*\rangle\right] = \langle \nabla J(\boldsymbol{\vartheta}), \boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\rangle + \mathbb{E}\left[z' H_{\boldsymbol{\vartheta}} z\right] \\ &= \langle \nabla J(\boldsymbol{\vartheta}), \boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\rangle + \sigma^2 \operatorname{tr}(\boldsymbol{H_\vartheta})\end{aligned} \tag{4.4}$$

In a convex objective, Hessian matrix is positive semi-definite and $\operatorname{tr}(\boldsymbol{H_\vartheta})$ is the summation of the eigenvalues of $H_{\boldsymbol{\vartheta}}$. Additionally, the right-hand side of Eq (4.3) can further be express as follows:

$$\mathbb{E}\left[\left\|\tilde{\boldsymbol{\vartheta}} - \boldsymbol{\vartheta}_*\right\|^2\right] = \mathbb{E}\left[\|\boldsymbol{\vartheta} - z - \boldsymbol{\vartheta}_*\|^2\right] = \nu\left(\|\boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\|^2 + p\sigma^2\right) \tag{4.5}$$

We therefore estimate the value of ν as stated in Definition 5 based on the above approximation.

$$\nu \lesssim \frac{\operatorname{tr}(\boldsymbol{H_\vartheta})\,\sigma^2 + \boldsymbol{\mu}\,\|\boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\|^2}{p\sigma^2 + \|\boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\|^2}$$

In a relatively large σ^2 domain, it implies $\nu \approx \frac{\operatorname{tr}(\boldsymbol{H_\vartheta})}{p}$ that is the average curvature at $\boldsymbol{\vartheta}$. This large σ^2 is a practicable setting since significant DP guarantee demand non-trivial volumes of perturbed noise. The above assessment advocate that ν capable of been independent and much greater than μ. Making it undeniably valid for countless convex objectives. Considering l_2 regularized logistic regression. The objective function is strongly convex exclusively based on l_2 regularizer. Hence, minimum curvature i.e., regularization λ is the strongly convex coefficient.

4.3. Utility guarantee of our proposed iterative gradient perturbation with expected curvature architecture

We demonstrate the utility bound of our proposed Iterative Gradient Perturbation-based algorithm can be improved based on the expected curvature.

Algorithm 1: Iterative Gradient Perturbation with Expected Curvature Algorithm

```
Input: $J_t^i(\vartheta) := \ell\left(\vartheta, x_t^i, y_t^i\right), i \in [1,k]$ and $t \in [0,T]$; initial points $\vartheta_0^1,\ldots,\vartheta_0^m$; maximum
       iterations $T$;Privacy parameters $\epsilon,\delta$; learning rate $\eta$; regularization parameter $\lambda$; Loss
       function $\ell(\vartheta)$ with Lipschitz constant $L$
Output: The optimal model parameter $\vartheta_T$
Randomly initialize the model parameter $\vartheta_0$;
for $i = 0..., T$ do
    for each node $i = 1, ..., k$ do
        Compute;
        Compute gradient: $\boldsymbol{V}_t = \nabla J\left(\boldsymbol{\vartheta}_t\right)$;
        Update parameter: $\boldsymbol{\vartheta}_{t+1} = \boldsymbol{\vartheta}_t - \eta_t\left(\boldsymbol{V}_t + \boldsymbol{z}_t\right)$, where $\boldsymbol{z}_t \sim \mathcal{N}\left(0, \sigma_t^2 I_p\right)$
    end for
end for
```

Algorithm 1 is (ϵ, δ)-DP if we set $\sigma_t = \Theta\left(\frac{L\sqrt{T\log(1/\delta)}}{kn\epsilon}\right)$. Let$\{\vartheta_1, ..., \vartheta_T\}$ be the training path whiles $v = \min\{v_1, ..., v_T\}$ is the minimum expected curvature over the path. Furthermore, we demonstrate the utility guarantee of our proposed algorithm where v > 0.

Theorem 9. *Utility guarantee of our proposed Iterative Gradient Perturbation-based algorithm is achieved where* $v > 0$. *Let assume G is L-Lipschitz and β-smooth alongside v expected curvature. We therefore set $\sigma_t = \Theta(L\sqrt{T\log(1/\delta)}/kn\epsilon)$, learning rate $\eta \leq \frac{1}{\beta}$ and $T = \frac{2\log(n)}{\eta v}$ we have:*

$$\mathbb{E}\left[J\left(\boldsymbol{\vartheta}_{T+1}\right) - J\left(\boldsymbol{\vartheta}_*\right)\right] = O\left(\frac{\beta p \log(n) L^2 \log(1/\delta)}{v^2 k n^2 \epsilon^2}\right)$$

Proof. Let assume $\{\vartheta_1, \ldots, \vartheta_t\}$ is the path produced by the optimization approach. since ϑ_t holds Gaussian perturbation statistical noise z_{t-1}, with definition [12] we obtain:

$$\mathbb{E}_{z_{t-1}}\left[\left\langle \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*, \nabla J\left(\boldsymbol{\vartheta}_t\right)\right\rangle\right] \geq v_t \mathbb{E}_{z_{t-1}}\left[\left\|\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*\right\|^2\right]$$

since J is β -smooth, we obtain

$$\left\langle \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*, \nabla J\left(\boldsymbol{\vartheta}_t\right)\right\rangle \geq \frac{1}{\beta}\left\|\nabla J\left(\boldsymbol{\vartheta}_t\right)\right\|^2$$

we therefore take the linear combination of the above inequalities

$$\begin{aligned}\mathbb{E}_{z_{t-1}}[\langle \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*, \nabla J(\boldsymbol{\vartheta}_t)\rangle] &\geq \theta \nu_t \mathbb{E}_{z_{t-1}}[||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||^2] + \frac{(1-\theta)}{\beta}\mathbb{E}_{z_{t-1}}[||\nabla J(\boldsymbol{\vartheta}_t)||^2] \\ &\geq \theta \nu \mathbb{E}_{z_{t-1}}[||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||^2] + \frac{(1-\theta)}{\beta}\mathbb{E}_{z_{t-1}}[||\nabla J(\boldsymbol{\vartheta}_t)||^2]\end{aligned} \tag{4.6}$$

Let $r_t = ||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||$ at a resultant error for step t. We achieve the preceding inequalities connecting r_t and r_{t+1}

$$r_{t+1}^2 = ||\boldsymbol{\vartheta}_t - \eta \nabla J(\boldsymbol{x}_t) - \eta z_t - \boldsymbol{\vartheta}_*||^2 = ||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||^2 - 2\eta\langle \nabla J(\boldsymbol{\vartheta}_t) + \boldsymbol{z}_t, \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*\rangle + \eta^2||\nabla J(\boldsymbol{\vartheta}_t) + \boldsymbol{z}_t||^2 \tag{4.7}$$

Taking the expectation based on z_t, we obtain:

$$\mathbb{E}_{z_t}[r_{t+1}^2] \leq ||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||^2 - 2\eta\langle \nabla J(\boldsymbol{\vartheta}_t), \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*\rangle + \eta^2||\nabla J(\boldsymbol{\vartheta}_t)||^2 + p\eta^2\sigma_t^2 \tag{4.8}$$

Additionally, take expectation relative to z_{t-1} and use Eq (4.8), we now obtain

$$\begin{aligned}&\mathbb{E}_{z_t,z_{t-1}}[r_{t+1}^2] \leq \mathbb{E}_{z_{t-1}}[||\boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*||^2] - 2\eta\mathbb{E}_{z_{t-1}}[\langle \nabla J(\boldsymbol{\vartheta}_t), \boldsymbol{\vartheta}_t - \boldsymbol{\vartheta}_*\rangle] + \eta^2\mathbb{E}_{z_{t-1}}[||\nabla J(\boldsymbol{x}_t)||^2] + p\eta^2\sigma_t^2 \\ &\leq (1 - 2(1-\theta)\eta\nu)\mathbb{E}_{z_{t-1}}[r_t^2] + (\eta^2 - \frac{2\eta\theta}{\beta})\mathbb{E}_{z_{t-1}}[||\nabla J(\boldsymbol{\vartheta}_t)||^2] + p\eta^2\sigma_t^2\end{aligned} \tag{4.9}$$

We now let $\eta \leq \frac{1}{\beta}$, $\theta = \frac{1}{2}$

$$\mathbb{E}_{z_t,z_{t-1}}[r_{t+1}^2] \leq (1-\eta\nu)\mathbb{E}_{z_{t-1}}[r_t^2] + p\eta^2\sigma_t^2 \tag{4.10}$$

With the application of Eq (4.10) and taking expectation with respect to $z_t, z_{t-1}, \cdots, z_1$ iteratively yields:

$$\mathbb{E}[r_{t+1}^2] \leq (1-\eta\nu)^t r_1^2 + p\eta^2\sum_{i=1}^{t}(1-\eta\nu)^{t-i}\sigma_i^2 \tag{4.11}$$

The uniform privacy budget allocation scheme is set to:

$$\sigma_t^2 = \Theta\left(\frac{TG^2\log(1/\delta)}{kn^2\epsilon^2}\right)$$

Therefore

$$\mathbb{E}\left[r_{T+1}^2\right] \leq (1-\eta\nu)^T r_1^2 + \Theta\left(\frac{p\eta TG^2\log(1/\delta)}{\nu kn^2\epsilon^2}\right) \tag{4.12}$$

Let $T \geq \frac{2\log(n)}{\eta\nu}$, we have

$$\begin{aligned}(1-\eta\nu)^T r_1^2 &= \exp\left(\frac{\log(1-\eta\nu)\log\left(n^2\right)}{\eta\nu}\right)r_1^2 = \exp\left(\log\left(1/n^2\right)\frac{1}{\eta\nu}\log\left(1+\frac{\eta\nu}{1-\eta\nu}\right)\right)r_1^2 \\ &\leq \left(\frac{1}{kn^2}\right)^{\frac{1}{\eta\nu}\log\left(1+\frac{\eta\nu}{1-\eta\nu}\right)} r_1^2 < \frac{r_1^2}{kn^2}\end{aligned} \tag{4.13}$$

The final inequality holds because $\frac{1}{\eta\nu}\log\left(1+\frac{\eta\nu}{1-\eta\nu}\right) > 1$ for $\frac{1}{\eta\nu} \geq \frac{\beta}{\nu} \geq 1$ Therefore, for $T \geq \frac{2\log(n)}{\eta\nu}$, we obtain the excepted solution error $\mathbb{E}\left[r_{T+1}^2\right]$ satisfies

$$\mathbb{E}\left[r_{T+1}^2\right] = O\left(\frac{p\eta T L^2 \log(1/\delta)}{\nu k n^2 \epsilon^2}\right) \tag{4.14}$$

since $J(x)$ is β-smooth, we obtain

$$J(\boldsymbol{\vartheta}) - J(\boldsymbol{\vartheta}_*) \leq \frac{\beta}{2}\|\boldsymbol{\vartheta} - \boldsymbol{\vartheta}_*\|^2 \tag{4.15}$$

Using Eqs (4.14) and (4.15) we have the excepted excess risk satisfies

$$\mathbb{E}\left[J(\boldsymbol{\vartheta}_{T+1}) - J(\boldsymbol{\vartheta}_*)\right] = O\left(\frac{\beta p\eta T G^2 \log(1/\delta)}{\nu k n^2 \epsilon^2}\right)$$

when $T \geq \frac{2\log(n)}{\eta\nu}$. In this setting we minimized the utility bound where $T = \frac{2\log(n)}{\eta\nu}$.

5. Performance analysis

This section validates the productivity of our algorithm based on both classification and logistic regression on four (4) real-world data sets: (a) CICMalDroid2020 [38] dataset is a composition of current samples of five (5) different apps category: Benign, Banking, Riskware, Adware and SMS containing 17,341 instances. (b) CICIDS2018 [39] dataset, a composition of 16,000,000 intrusion detection dataset covering a wide range of attack types. (c) Adult [40] as a US 1994 Census data set contains 48,842 records of citizens(d) IPUMS-US data contains Census data obtained from IPUMS-International [41]. We compare our proposed algorithm with the differential privacy output perturbation and gradient perturbation in [36] for logistic regression with L_2-norm regularization.

Table 2. Summary of the datasets.

Dataset	Size (n)	Dimension
CICMalDroid2020 [38]	17,341	470
CICIDS2018 [39]	16,000,000	125
Adult [40]	48,842	124
IPUMS-US [41]	40,000	58

5.1. Benchmark for comparison

In this domain, it is important to predict if the interconnection is a denial-of-service (DoS) attack or otherwise. We indiscriminately sample 70,000 individual data along with dividing amongst training dataset of 50,000 records whiles the 20,000 records are used as a test set. With $\mathbf{x}_i \in \mathbb{R}^{p+1}, y_i \in \{-1, +1\}$, and $\lambda > 0$ as the regularization coefficient. Throughout our simulations; we set coefficient $\lambda = 0.001$, learning rate $\eta = 1$, failure probability $\theta = 0.001$, $\epsilon = 0.05$ privacy budget, G-Lipschitz constant $= 1$ and entire iterations $T = 1000$ for GD. We validate our proposed output perturbation and gradient perturbation-based algorithms with other benchmarks relative to optimality and relative accuracy loss. In the case of regression, relative accuracy loss is termed as mean square error (MSE) θ and θ^* which is the difference in accuracy over the test data.

We also explore the performance of our proposed algorithm's output perturbation and iterative gradient perturbation-based algorithm with benchmarks based on accuracy loss and relative optimality gap. Optimality gap measures empirical risk bound $\mathcal{J}(\theta) - \mathcal{J}(\theta^*)$ of the training dataset, θ is the optimum non-private classifier in the centralized domain. Nonetheless, relative accuracy loss becomes the variance in the accuracy (i.e., MSE in regression) of θ and θ^* over text data set. Relative accuracy loss and optimality gap of the entire model is measured up to 1500 training iterations for the gradient descent whiles reporting the outcome for diverse partitioned datasets. Dataset owners k are varied from 100 participants with each one of them possessing 500 instances of data up to 1000 participants with each one of them also containing 50 dataset instances and up to 50,000 participants each containing only one dataset instance. with $\mathbf{x}_i \in \mathbb{R}^{p+1}, y_i \in \{-1, +1\}$, and $\lambda > 0$ as the regularization coefficient.

In our proposed output perturbation-based model aggregation (MPC-OP), we demonstrate the comparison of this model with Pathak et al. [36] which is represented as (PAT), other cutting-edge differential privacy benchmarks are also achieved by the application of objective perturbation and output perturbation techniques of Wang et al. [33] on each of the locally trained model estimator $\widehat{\theta_j}$ to attain a differentially private local model estimator whiles aggregation of the classifier is computed to attain differentially private aggregated classifier $\widehat{\theta}_{priv}$ with confidence intervals for the parameters in the model. For our experiments on our proposed adaptive iterative gradient perturbation-based learning algorithm, we adopt a benchmark of aggregation for locally perturbed gradients [42] with the aim of improving the noise bound by applying zCDP and coupling strategy [43] also for the verification of the privacy budget. Our proposed output perturbation-based model aggregation and adaptive iterative gradient perturbation-based frameworks in the multi-party setting is represented as MPC-OP and MPC-AIGP respectively. In all our simulations there is a variation of the number of data owners p from 100–1000 with up to 50,000 data owners with each of them possessing only one data instance.

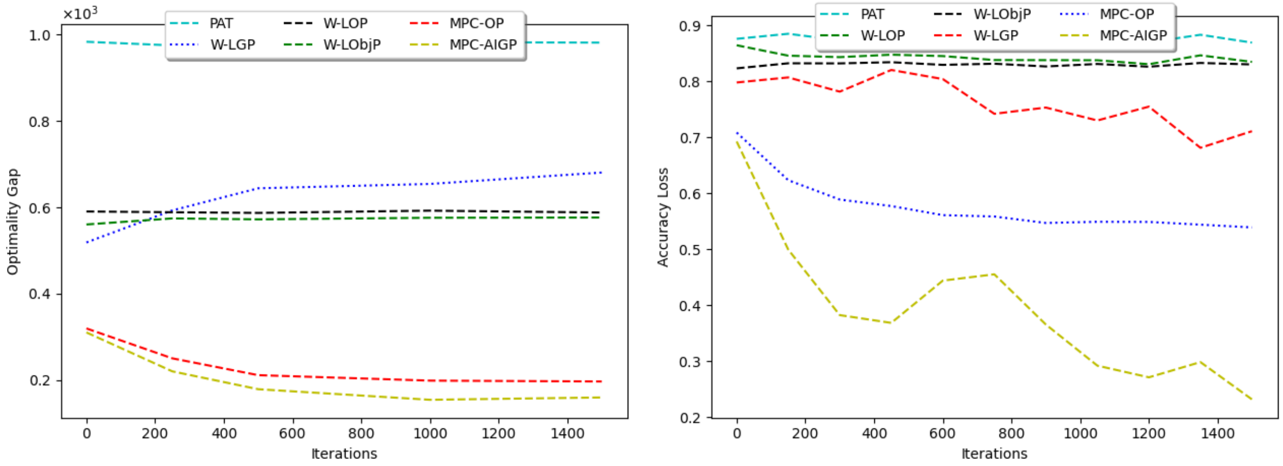

Figure 3. Relative accuracy loss and optimality gap comparison on adult (p = 1000). All models have privacy budget of $\epsilon = 0.05$ for each iteration.

Result on adult dataset: The adult [40] data is a composition of demographic information of approximately 47,000 individuals, In this domain, our duty is to predict if annual income of data owners exceed or below \$50,000.00 threshold. During the pre-processing stage, we obtained 104 features for each of the records, whiles the missing values were removed therefore yielding 45,222 records with 30,000 of them now forming the training dataset whiles the rest are used for the testing of our model. In our simulation on the Adult dataset, our proposed MPC-OP and MPC-AIGP methods outperform the benchmarks both with respect to the accuracy loss and optimality gab as demonstrated in Figure 4.

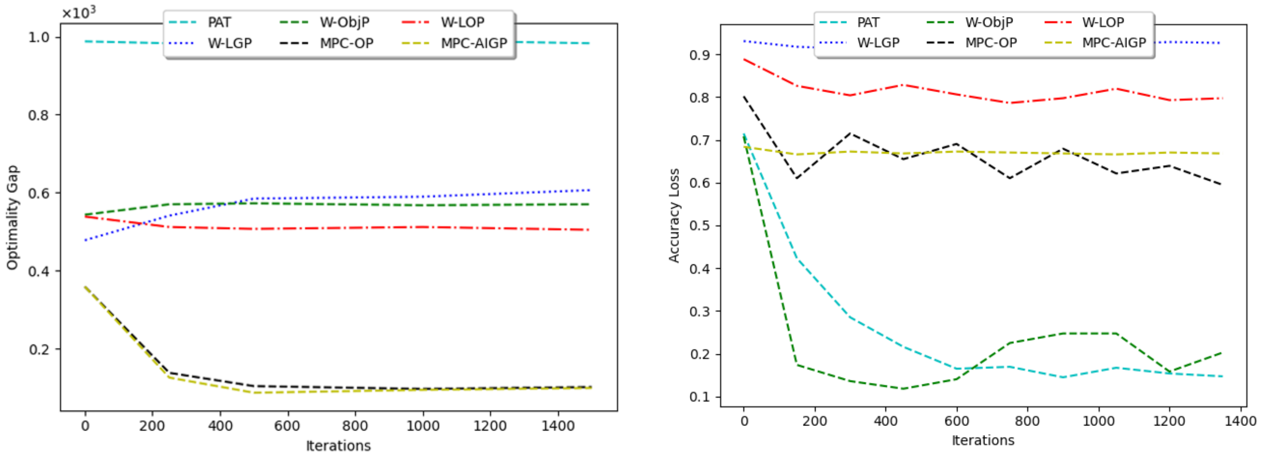

Figure 4. Relative accuracy loss and optimality gap comparison on CICMalDroid2020 (p = 1000). Entire models contains privacy budget of ϵ = 0:05 for each iteration.

Result on CICMalDroid2020 dataset: As the amount of data owners p grows and with the decrease in the size of the local data, the relative accuracy of all the models begins to decrease with the exclusion of MPC-AIGP as represented in Figure 4. It is obvious that the performance of the benchmarks algorithms begins to depreciate basically owing to the huge volumes of statistical noise injected within the classifier. Furthermore, the performance of MPC-OP also deteriorates with the reduction in the size of the local dataset owing to the loss in information from partitioning of the dataset which has been one of the challenges with the aggregation of locally trained classifiers.

Result on CICIDS2018 dataset: With the reduction in the amount of local dataset which is as a result of the decrease in the number of data owners p, there is a great decrease in the performance of all the methods with the exception of MPC-AIGP (Figure 5). Although there is a reduction in the performance of MPC-OP as the size of the local dataset is reduced, it continues to outperform the benchmarks of existing model aggregation. We observed that as $p = 1000$, the performance of W-LObjP is weaker as compared to the W-LOP, this is due to the deviation in the objective function of W-LObjP (n). It is important to note that the utility of PAT is greatly affected as a result of the huge amount of statistical noise injected into the model, resulting in the plot been out of range for $p = 1000$ (Figure 5).

5.2. *Analysis of expected curvature on our iterative gradient perturbation method*

The analysis in Eq (4.4) suggests that ν is capable of being independent and much larger than μ. In considering an instance in regularized logistic regression. The objective function becomes strongly

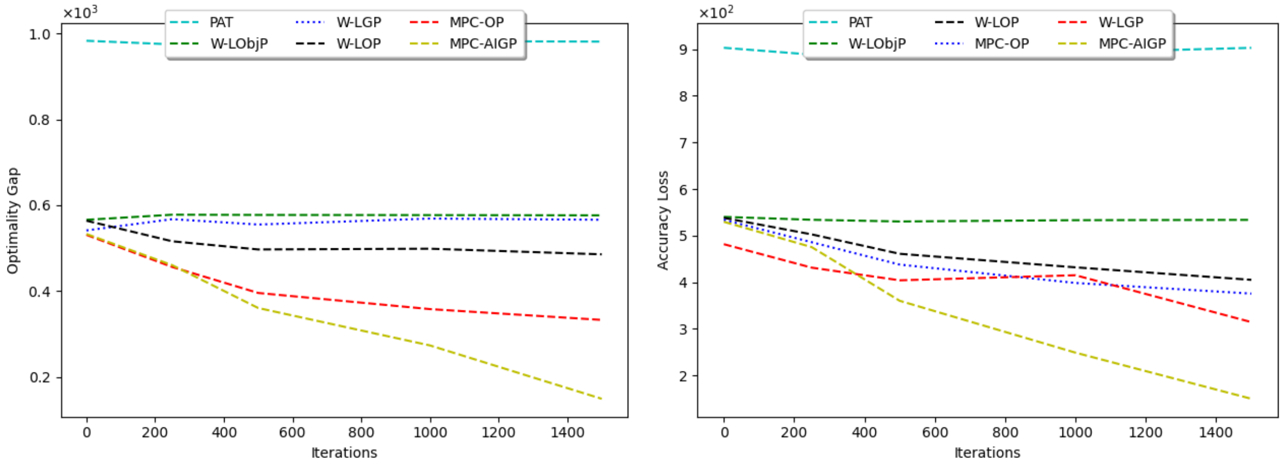

Figure 5. Relative accuracy loss and optimality gap comparison on CICIDS2018 (p = 1000). All models have privacy budget of $\epsilon = 0.05$ for each iteration.

convex based on the l_2 regularizer. Subsequently, making the minimum curvature (i.e., strongly convex coefficient λ) the regularization coefficient. We compare the average and minimum curvatures of regularized logistic regression throughout the learning process in Figure 6 and Figure 7. It is important to note that the average curvature is predominantly not affected by the λ which is the regularization term. Conversely, in some few first steps, the minimum curvature is able to reach λ. Consequently, the removal of the independence on the minimum curvature has become a substantial improvement. As we plot the curvature of CICMalDroid2020 dataset in Figure 6 and IPUMS dataset in Figure 7, it was obvious that the resulting curvatures is similar.

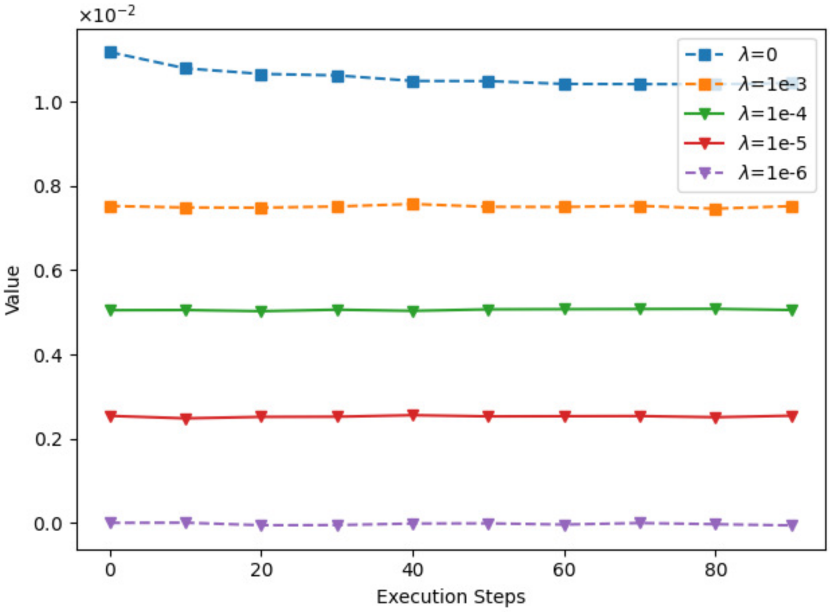

Figure 6. Curvature of regularized logistic regression on CICMalDroid2020 dataset over training. With the square dotted symbols and down triangles representing average and minimum curvature respectively.

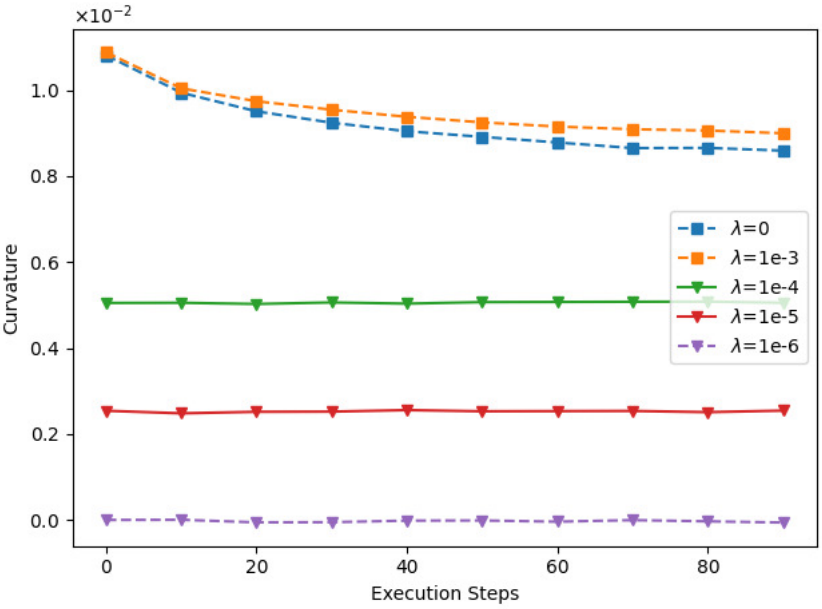

Figure 7. Curvatures for regularized logistic regression trained on IPUMS dataset. The square dotted symbols in the plot represents average curvature and the down triangles represents minimum curvature.

Experimental results with variations in parameters: It is worth remembering that all real-world datasets used in our simulations contains bother numerical and categorical features. We therefore apply some of the common pre-processing computations in machine learning; by transforming all categorical features into a set of binary variables by the creation of one binary variable for the individual distinct class; every numerical feature is re-scaled into the range of [0, 1] to enforce equal scale for the features. We normalized the individual observation to a unit norm ($i.e., \| \ x_i \ \|_2 = 1 fori = 1, 2, ..., n$) to meet its specification. To demonstrate the effectiveness of our method on real-world data set by comparing it to other state-of-the-art algorithms. We apply it to a regularized logistic regression model with the following aim:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i \mathbf{w}^\top \mathbf{x}_i\right)\right) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

To be more precise, we also consider (regularized) logistic regression on the four (4) real world data sets. We therefore validate the minimization error $\mathbb{E}F\left(w_{privacy}, S\right) - F(\hat{w}, S)$ and running time of our algorithms based on different $\epsilon = \{0.05, 0.5, 0.1\}$ and $\delta = 0.001$ (see Table 3 for more details)

6. Conclusions

In this work, we establish that the injection of privacy noise actually improves utility guarantee of gradient descent optimization analysis, which can be reduced when the noise is generated and added within a secure computation in the distributed machine learning domain. The application of our output perturbation for the aggregation of our locally trained models attains ϵ-differential privacy. Our approach of secure aggregation applying secure multi-party computation practically enforces the

Table 3. Details of our simulation results.

Dataset	ϵ	δ	Error Our Algorithm (ϵ, δ)	[4](ϵ, δ)	Run-time Our Algorithm (ϵ, δ)	[4](ϵ, δ)
CICMalDroid2020	0.05	0.001	0.1714	1.3545	10.214	532.12
	0.5		0.2231	1.4585	36.796	519.33
	0.1		0.3983	2.2552	12.613	518.67
CICIDS2018	0.05	0.001	0.0092	0.3039	11.036	185.32
	0.5		0.0101	0.4162	13.893	190.87
	0.1		0.0292	0.4202	4.977	190.32
Adult	0.05	0.001	0.0912	0.3692	10.896	202.12
	0.5		0.0212	0.6082	68.504	253.72
	0.1		0.0429	0.6227	22.814	255.12
IPUMS	0.05	0.001	0.1925	3.2011	0.1811	6.3952
	0.5		4.0412	4.0090	0.4375	6.4452
	0.1		0.0564	5.45659	0.1431	6.5021

models whose inputs are encrypted with possibly distinct encryption schemes or even distinct keys is general enough to assist any machine learning method, and requires only a single secure model aggregation. The pivot of the proposed algorithm is to enhance the comprehension of the utility-privacy in DML, and offers mechanisms for increasing utility to attain reasonable privacy guarantee. The gradient perturbation algorithms also present (ϵ, δ)-differential privacy and also theoretically justify its empirical superiority in our proposed algorithm over other existing algorithms. The gradient perturbation method proceeds to gain grounds in cutting-edge utility guarantee of DP-ERM algorithm. Performance evaluation on real-world human recognition activity datasets establish that our protocol incurs minimal computational overhead, provide substantial utility gains for typical security and privacy guarantees. Our experiment on these datasets also accurately verifies our theoretical findings.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No.61672135), the Frontier Science and Technology Innovation Projects of National Key R&D Program (No.2019QY1405), the Sichuan Science and Technology Innovation Platform and Talent Plan (No.20JCQN0256), and the fundamental Research Funds for the Central Universities (No.2672018ZYGX2018J057).

Conflict of interest

All authors declare that there is no conflicts of interest in this paper.

References

1. Y. Chen, Y. Mao, H. Liang, S. Yu, Y. Wei, S. Leng, Data poison detection schemes for distributed machine learning, *IEEE Access*, **8** (2019), 7442–7454.
2. C. Dwork, G. N. Rothblum, S. P. Vadhan, Boosting and differential privacy, in 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, (2010), 51–60.
3. L. Wang, Q. Gu, Differentially private iterative gradient hard thresholding for sparse learning, in *28th International Joint Conference on Artificial Intelligence*, 2019.
4. B. Jayaraman, L. Wang, D. Evans, Q. Gu, Distributed learning without distress: Privacy-preserving empirical risk minimization, *Adv. Neural Inf. Process. Syst.*, 2018.
5. Y. Xu, G. Yang, S. Bai, Laplace input and output perturbation for differentially private principal components analysis, *Secur. Commun. Networks*, **2019** (2019).
6. D. Yu, H. Zhang, W. Chen, J. Yin, T. Liu, Gradient perturbation is underrated for differentially private convex optimization, preprint, arXiv:1911.11363.
7. A. G. Thakurta, *Differentially Private Convex Optimization For Empirical Risk Minimization And High-dimensional Regression*, Ph.D thesis, The Pennsylvania State University, 2012.
8. X. Ma, C. Ji, X. Zhang, J. Wang, J. Li, K. Li, Secure multiparty learning from the aggregation of locally trained models, *J. Network Comput. Appl.*, **167** (2020), 102754.
9. O. Kwabena, Z. Qin, T. Zhuang, Z. Qin, Mscryptonet: Multi-scheme privacy-preserving deep learning in cloud computing, *IEEE Access*, **7** (2019), 29344–29354.
10. A. Gascón, P. Schoppmann, B. Balle, M. Raykova, J. Doerner, S. Zahur,et al., Privacy-preserving distributed linear regression on high-dimensional data, *PoPETs*, **4** (2017), 345–364.
11. X. Wang, S. Ranellucci, J. Katz, Global-scale secure multiparty computation, in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, (2017), 39–56.
12. C. Dwork, A. Roth, The algorithmic foundations of differential privacy, *Found. Trends Theor. Comput. Sci.*, **9** (2014), 211–407.
13. S. Mahloujifar, M. Mahmoody, A. Mohammed, Data poisoning attacks in multi-party learning, in *International Conference on Machine Learning*, PMLR, (2019), 4274-4283.
14. E. Kim, H. Lee, J. Park, Towards round-optimal secure multiparty computations: Multikey FHE without a CRS, *Int. J. Found. Comput. Sci.*, **31** (2020), 157–174.
15. D. Croce, F. Giuliano, I. Tinnirello, L. Giarré, Privacy-preserving overgrid: Secure data collection for the smart grid, *Sensors*, **20** (2020), 2249.
16. J. Liu, Y. Tian, Y. Zhou, Y. Xiao, N. Ansari, Privacy preserving distributed data mining based on secure multi-party computation, *Comput. Commun.*,**153** (2020), 208–216.
17. O. Catrina, C. Dragulin, Multiparty computation of fixed-point multiplication and reciprocal, in *2009 20th International Workshop on Database and Expert Systems Application*, (2009),107–111.
18. O. Catrina, A. Saxena, Secure computation with fixed-point numbers, in *International Conference on Financial Cryptography and Data Security*, Springer, Berlin, Heidelberg, (2010), 35–50.

19. A. C. Yao, Protocols for secure computations, in *23rd annual symposium on foundations of computer science (sfcs 1982)*, (1982), 160–164.

20. S. Goldwasser, Multi party computations: past and present, in *Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing*, (1997), 1–6.

21. C. Gentry, Fully homomorphic encryption using ideal lattices, in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, (2009), 169–178.

22. I. Damgård, C. Orlandi, Multiparty computation for dishonest majority: From passive to active security at low cost, in *Annual cryptology conference*, Springer, Berlin, Heidelberg, (2010), 558–576.

23. R. Bendlin, I. Damgård, C. Orlandi, S. Zakarias, Semi-homomorphic encryption and multiparty computation, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, (2011), 169–188.

24. J. B. Nielsen, P. S. Nordholt, C. Orlandi, S. S. Burra, A new approach to practical active-secure two-party computation, in *Annual Cryptology Conference*, Springer, Berlin, Heidelberg, (2012), 681–700.

25. A. Bansal, T. Chen, S. Zhong, Privacy preserving back-propagation neural network learning over arbitrarily partitioned data, *Neural Comput. Appl.*, **20** (2011), 143–150.

26. J. Yuan, S. Yu, Privacy preserving back-propagation neural network learning made practical with cloud computing, *IEEE Trans. Parallel Distrib. Syst.*, **25** (2014), 212–221.

27. W. Zhang, A BGN-type multiuser homomorphic encryption scheme, in *2015 International Conference on Intelligent Networking and Collaborative Systems*, IEEE, (2015), 268–271.

28. E. Hesamifard, H. Takabi, M. Ghasemi, C. Jones, Privacy-preserving machine learning in cloud, in *Proceedings of the 2017 on cloud computing security workshop*, (2017), 39–43.

29. P. Li, J. Li, Z. Huang, T. Li, C. Gao, S. Yiu, et al., Multi-key privacy-preserving deep learning in cloud computing, *Future Gener. Comput. Syst.*, **74** (2017), 76–85.

30. P. Mukherjee, D. Wichs, Two round multiparty computation via multi-key FHE, in M. Fischlin and J. Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, (2016), 735–763.

31. R. Agrawal, R. Srikant, Privacy-preserving data mining, in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, (2000), 439–450.

32. P. K. Fong, J. H. Weber-Jahnke, Privacy preserving decision tree learning using unrealized data sets, *IEEE Trans. Knowl. Data Eng.*, **24** (2012), 353–364.

33. Y. Wang, D. Kifer, J. Lee, Differentially private confidence intervals for empirical risk minimization, *J. Priv. Confidentiality*, **9** (2019).

34. M. Bun, T. Steinke, Concentrated differential privacy: Simplifications, extensions, and lower bounds, in *Theory of Cryptography Conference*, Springer, Berlin, Heidelberg, (2016), 635–658.

35. W. Du, A. Li, Q. Li, Privacy-preserving multiparty learning for logistic regression, in *International Conference on Security and Privacy in Communication Systems*, Springer, Cham, (2018), 549–568.

36. M. A. Pathak, S. Rane, B. Raj, Multiparty differential privacy via aggregation of locally trained classifiers, in *NIPS*, (2010), 1876–1884.

37. K. Chaudhuri, C. Monteleoni, Privacy-preserving logistic regression, in *NIPS*, **8** (2008), 289–296.

38. S. Mahdavifar, A. F. A. Kadir, R. Fatemi, D. Alhadidi, A. A. Ghorbani, Dynamic android malware category classification using semi-supervised deep learning, in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, (2020), 515–522.

39. I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in *ICISSp*, (2018), 108–116.

40. M. Lichman, *UCI machine learning repository*, 2013. Available from: http://archive. ics. uci. edu/ml.

41. Minnesota Population Center, Integrated Public Use Microdata Series, International: Version 6.4, 2015.

42. R. Shokri, V. Shmatikov, Privacy-preserving deep learning, in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, (2015), 1310–1321.

43. A. Albarghouthi, J. Hsu, Synthesizing coupling proofs of differential privacy, *Proc. ACM Program. Lang.*,**2** (2017), 1–30.

AIMS Press