

Towards Distributed Privacy-Preserving Prediction

1st Lingjuan Lyu
Department of Computer Science
National University of Singapore
lyulj@comp.nus.edu.sg

2nd Yee Wei Law
School of Engineering
University of South Australia
yeewei.law@unisa.edu.au

3rd Kee Siong Ng
Software Innovation Institute
Australian National University
keesiong.ng@anu.edu.au

4th Shibe Xue
Department of Automation,
Shanghai Jiao Tong University
Key Laboratory of System Control
and Information Processing,
Ministry of Education of China
shbxue@sjtu.edu.cn

5th Jun Zhao, 6th Mengmeng Yang
School of Computer Science and Engineering
Nanyang Technological University
Singapore
junzhao, melody.yang@ntu.edu.sg

7th Lei Liu
Unicloud Engine Technology Co., Ltd
China
liulei@unicde.com

Abstract—In privacy-preserving machine learning, individual parties are reluctant to share their sensitive training data due to privacy concerns. Even the trained model parameters or prediction can pose serious privacy leakage. To address these problems, we demonstrate a generally applicable *Distributed Privacy-Preserving Prediction* (DPPP) framework, in which instead of sharing more sensitive data or model parameters, an untrusted aggregator combines only multiple models' predictions under provable privacy guarantee. Our framework integrates two main techniques to guarantee individual privacy. First, we introduce the improved Binomial Mechanism and Discrete Gaussian Mechanism to achieve distributed differential privacy. Second, we utilize homomorphic encryption to ensure that the aggregator learns nothing but the noisy aggregated prediction. Experimental results demonstrate that our framework has comparable performance to the non-private frameworks and delivers better results than the local differentially private framework and standalone framework.

Index Terms—Privacy-Preserving, prediction, distributed differential privacy, homomorphic encryption.

I. INTRODUCTION

Many real-world applications would benefit from collaborative learning among multiple parties. This trend is motivated by the fact that the data owned by a single party may be very heterogeneous, resulting in an overfit model that might deliver inaccurate results when applied to other data. On the other hand, there is much demand to perform machine learning in a collaborative manner, since massive amount of data are often required to ensure sufficient computational power for test purpose. However, the increasing privacy and confidentiality concerns pose obstacles to collaboration [1]. For the sake of privacy, most approaches cannot afford to share the trained models publicly. Even the prediction output by a trained model can reveal training data privacy through black-box attacks [2]. Therefore, neither training data, trained model

nor model prediction should be directly shared. Meanwhile, these privacy concerns can be largely reduced if appropriate privacy-preserving schemes can be applied before the relevant statistic is released.

To mitigate privacy concerns in the distributed setting, instead of sharing more sensitive local data or model parameters of each party, we examine an alternative approach called *distributed privacy-preserving prediction* (DPPP), which allows parties to keep full control of their local data, and only share local model predictions in a privacy-preserving manner. In our approach, each party first trains a local model based on local training data \mathbf{D}_i . To answer the prediction query for any test point x , each party takes local model as the prediction function f to predict x , which returns the votes for all c classes, *i.e.*, $f(x, \mathbf{D}_i) = \mathbf{y}_i$, where $\mathbf{y}_i \in \{0, 1\}^c$ is an one-hot prediction vector that sums up to 1. Considering individual privacy, we appropriately perturb model predictions before releasing them for aggregation. The primitive of this provably private noisy sum can be referred to [3]. **Our contributions** include:

- We formulate a distributed privacy-preserving prediction framework, named DPPP, which combines *distributed differential privacy* (DDP) and homomorphic encryption to ensure individual privacy, maintain utility and provide aggregator obliviousness, *i.e.*, the aggregator learns nothing but the noisy aggregated prediction.
- We explore the stability of *Binomial Mechanism* (BM) and *Discrete Gaussian Mechanism* (DGM) to guarantee (ϵ, δ) -differential privacy in the distributed setting, and formally provide a tightest bound to date for BM.
- Extensive experiments demonstrate that DPPP delivers comparable performance to the non-private frameworks, and yields better results than the local differentially private and standalone frameworks.

Corresponding to: shbxue@sjtu.edu.cn. This work was supported in part by the National Natural Science Foundation of China under Grants 61873162 and in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No.ICT20052).

II. PRELIMINARIES AND RELATED WORK

A. Distributed Differential Privacy

Definition 1 ($(\epsilon, \delta, \gamma)$ -Distributed DP [4]). Let $\epsilon > 0$, $0 \leq \delta < 1$ and $0 < \gamma < 1$. We say that the mechanism \mathcal{M} with randomness over the joint distribution of $\mathbf{r} := (\mathbf{r}_1, \dots, \mathbf{r}_N)$ preserves $(\epsilon, \delta, \gamma)$ -distributed differential privacy (DDP) if the following conditions hold: For any neighbouring databases $D, D' \in \mathcal{D}^N$ that differ in one record, for any measurable subset $S \subseteq \mathcal{R}$, and for any subset \bar{K} of at least γN honest parties,

$$\Pr\{\mathcal{M}(D) \in S | \mathbf{r}_{\bar{K}}\} \leq \exp(\epsilon) \cdot \Pr\{\mathcal{M}(D') \in S | \mathbf{r}_{\bar{K}}\} + \delta.$$

In the above definition, γ is the fraction of uncompromised parties, and the probability is conditioned on the randomness $\mathbf{r}_{\bar{K}}$ from compromised parties, *i.e.*, it ensures that if at least γN participants are honest and uncompromised, we will accumulate noise of a similar magnitude as that of the *central differential privacy* (CDP) [5]. For differentially-private aggregation of local statistics, DDP permits each party to randomise its local statistic to a lesser degree than *local differential privacy* (LDP) [6], [7]. The most recent work introduced amplification by shuffling to lower the privacy cost of LDP algorithm when viewed in the central model of DP [8], [9]. LDP with shuffling yields a trust model which sits in between the central and local models for DP.

B. Homomorphic Encryption

Additive homomorphic encryption allows the calculation of the encrypted sum of plaintexts from their corresponding ciphertexts. Although there are several additive homomorphic cryptographic schemes, we use the threshold variant of Paillier scheme [10] in our framework, because it not only allows additive homomorphic encryption, but also distributes decryption among parties. In this cryptosystem, a party can encrypt the plaintext $m \in \mathbb{Z}_n$ with the public key $pk = (g, n)$ as

$$c = E_{pk}(m) = g^{m \cdot r^n} \mod n^2, \quad (1)$$

where $r \in \mathbb{Z}_n^*$ (\mathbb{Z}_n^* denotes the multiplicative group of invertible elements of \mathbb{Z}_n) is selected randomly and privately by each party. The additive homomorphic property of this cryptosystem can be described as:

$$\begin{aligned} E_{pk}(m_1 + m_2) &= E_{pk}(m_1) \cdot E_{pk}(m_2) \\ &= g^{m_1 + m_2} (r_1 r_2)^n \mod n^2, \end{aligned} \quad (2)$$

where m_1, m_2 are the plaintexts that need to be encrypted, and r_1, r_2 are the private randoms.

In this paper, (N, t) -threshold Paillier cryptosystem is adopted, in which the private key sk is distributed among N parties (denoted as $\{sk_1, sk_2, \dots, sk_N\}$), thus no single party has the complete private key. For any ciphertext c , each party i ($1 \leq i \leq N$) computes a partial decryption with its own partial private key sk_i as:

$$c_i = c^{2^{N!sk_i}} \quad (3)$$

Then based on the combining algorithm in [10], at least t partial decryptions are required to recover the plaintext m .

C. Multi-party Privacy

In multi-party scenario where data is sourced from multiple parties and the server is **not trustworthy**, individual privacy has to be protected. Without homomorphic encryption, each party has to add sufficient noise to their statistics before sending them to a central server to ensure LDP [7]. Since the aggregation sums up individual noise shares, the aggregated noise might render the aggregation useless. To preserve privacy without significantly degrading utility, differential privacy can be made distributed by combining with cryptographic protocols, as evidenced in [4], [6], [11], [12].

More recently, Agarwal *et al.* [13] recalled the Binomial Mechanism [14] and provided a similar bound as ours in Theorem 1. However, they focus on the privacy of the gradients aggregated from clients in federated learning, which is different from the problem studied in this work. More importantly, they did not offer a complete scheme to protect against the untrusted aggregator. Note that they did not provide a complete proof. Instead, we provide a detailed proof for the tight bound.

Another recent work is *Private Aggregation of Teacher Ensembles* (PATE) proposed by Papernot *et al.* [15]. PATE first trains an ensemble of teachers on disjoint subsets of private data. These teachers are then used to train a student model that can accurately mimic the ensemble. However, PATE assumes a trusted aggregator, who counts teacher votes assigned to each class, adds carefully calibrated Laplace noise to the resulting vote histogram, and outputs the class with the most noisy votes as the ensemble's prediction. Therefore, PATE fails to take into consideration against a potentially untrusted aggregator.

III. PROBLEM DEFINITION

Similar to Shi *et al.* [4], we consider an untrusted aggregator who may have arbitrary auxiliary information. For example, the aggregator may collude with a set of compromised parties, who can *reveal their data and noise values* to the aggregator as a form of auxiliary information. Our goal is to guarantee the privacy of each individual against an untrusted aggregator, even when the aggregator has arbitrary auxiliary information. To achieve this goal, we blind and encrypt the local statistics of parties before sharing them with the aggregator. Moreover, like most of the previous works [4], [11], [12], to ensure the correctness and functionality of the system, we do not consider a malicious aggregator, as it may not be desirable in many practical settings, and are not in the commercial interest of collaborative service providers for prediction service. We remark that our privacy model is stronger than [4] in the sense that we allow for party failures. This assumption is often more realistic, as one or more parties may fail to upload their encrypted values or fail to respond.

Moreover, we assume fewer than $1 - \gamma = 1/3$ of teachers are compromised – the rest are assumed to be honest. Decryption can be done by the remaining $2/3$ teachers using threshold

Paillier. Since cryptographic protocol requires discrete inputs, instead of adding floating-point Gaussian noise to each individual's prediction, we leverage Binomial Mechanism (BM) and Discrete Gaussian Mechanism (DGM) to generate discrete Binomial noise and Gaussian noise.

IV. DDP MECHANISMS

Approaches to DDP that implement an overall additive noise mechanism by summing the same mechanism run at each party (typically with less noise) necessitates mechanisms with stable distributions—to guarantee proper calibration of known end-to-end response distribution—and cryptography for hiding all but the final result from participating parties [4], [11], [12], [14]. DDP utilizes this nice stability to permit each party to randomise its local statistic to a lesser degree ($\frac{\sigma}{\sqrt{n}}$) than would LDP (σ) [6]. In summary, the goal of DDP is to both avoid the trust on any third party (trusted server in CDP [5] and trusted shuffler in LDP with shuffling [7]), and achieve better utility than LDP. On the other hand, if the server colludes with all the parties except the victim, the privacy guarantee would downgrade to LDP. We next introduce two representative stable distributions, including Binomial distribution and discrete Gaussian distribution, which can be seamlessly combined with cryptographic techniques.

A. Binomial Mechanism

Binomial Mechanism (BM) is based on the Binomial distribution $B(n, p)$ parameterized by n, p , where $n \in \mathbb{N}$ is the number of tosses, and $p \in (0, 1)$ is the success probability. We now define BM for the prediction function f with an output space in $\{0, 1\}^c$, where c is the number of classes, i.e., for each data point, f returns an one-hot prediction vector with a total of c elements in $\{0, 1\}$ that sum up to 1. Consider party i 's database \mathbf{D}_i and prediction f : let $f(\mathbf{x}, \mathbf{D}_i) = \mathbf{y}_i$ be the local prediction vector produced by party i given data \mathbf{D}_i , and y_i^j be the j -th element of \mathbf{y}_i , i.e., prediction (vote status) for class j . If party i assigns class j to input \mathbf{x} , then $y_i^j = 1$ while other elements are all 0's. The noisy vote count τ on each class j equals $\tau = \sum_{i=1}^N y_i^j + \text{noise}$, replacing the whole database \mathbf{D} with \mathbf{D}' differing only in one row changes the summation in each class by at most 1, i.e., sensitivity=1. Bounding the ratio of probabilities that τ occurs with inputs \mathbf{D} and \mathbf{D}' amounts to bounding the ratio of probabilities that **noise** = r^j and **noise** = $r^j + 1$, for different possible ranges of values of r^j . Given prediction function $f(\mathbf{x}, \mathbf{D}_i) = \mathbf{y}_i$, the goal of the BM is to compute the noisy vote count for each class (each coordinate of the aggregated prediction): $\sum_{i=1}^N y_i^j + r^j$, where $r^j = z - np$ is the random Binomial noise added to the vote count for each class j , and Binomial random variable $z \sim B(n, p)$ is independent for each class.

Theorem 1. (Tighter bound). For $p = 1/2$, Binomial Mechanism is (ϵ, δ) -differentially private so long as the total number of tosses $n \geq 2 \left(\frac{2+\epsilon}{\epsilon} \right)^2 \ln \left(\frac{2}{\delta} \right)$. Note that this lower bound is tighter than $n \geq 64 \ln \left(\frac{2}{\delta} \right) / \epsilon^2$ given in [14], but they both share the $\ln \left(\frac{2}{\delta} \right)$ term.

Proof. For Binomial distribution with $p = \frac{1}{2}$, $r = z - \frac{n}{2}$ is termed as Binomial noise, where z is a Binomial random variable sampled from $B(n, 1/2)$ with mean $\frac{n}{2}$ and success probability $\frac{1}{2}$ by performing coin flipping. To investigate how we can size Binomial noise, suppose r^j is the random Binomial noise added to the vote count for each class j , then

$$\tau = \sum_{i=1}^N y_i^j + r^j, \quad \tau' = \sum_{i=1}^N y_i^j + r^j + 1.$$

For the Binomial random variable with bias $1/2$, whose mass at $\frac{n}{2} + r^j$ is

$$\Pr \left(\frac{n}{2} + r^j \right) = \binom{n}{\frac{n}{2} + r^j} \left(\frac{1}{2} \right)^n,$$

ϵ -differential privacy requires that

$$\begin{aligned} \Pr \left\{ \frac{n}{2} + r^j \right\} &\leq e^\epsilon \Pr \left\{ \frac{n}{2} + r^j + 1 \right\} \\ \implies \binom{n}{\frac{n}{2} + r^j} \left(\frac{1}{2} \right)^n &\leq e^\epsilon \binom{n}{\frac{n}{2} + r^j + 1} \left(\frac{1}{2} \right)^n \\ \implies \frac{n}{2} + r^j + 1 &\leq e^\epsilon \left(\frac{n}{2} - r^j \right). \end{aligned}$$

To express r^j in terms of ϵ in an algebraically simple way, we use the inequality $1 + \epsilon \leq e^\epsilon$:

$$\frac{n}{2} + r^j + 1 \leq (1 + \epsilon) \left(\frac{n}{2} - r^j \right) \implies r^j \leq \frac{\epsilon n - 2}{4 + 2\epsilon} < \frac{\epsilon n}{4 + 2\epsilon}.$$

Therefore, the Binomial random variable $n/2 + r^j$ can *apparently* achieve ϵ -differential privacy, as long as $r^j < \frac{\epsilon n}{4 + 2\epsilon}$. Note:

- $\frac{n}{2} + \frac{\epsilon n}{4 + 2\epsilon} = \left(1 + \frac{\epsilon}{2 + \epsilon} \right) \frac{n}{2}$. This equation will be used in Eq. (4) later.
- This noise upper bound is tighter than Dwork *et al.*'s [14].

However, note that when r^j exceeds this upper bound, ϵ -differential privacy will be violated. Hence, we turn to the relaxed (ϵ, δ) -DP, which requires that

$$\Pr \left\{ \frac{n}{2} + r^j \leq \frac{n}{2} - \xi \text{ or } \frac{n}{2} + r^j \geq \frac{n}{2} + \xi \right\} \leq \delta,$$

where $\xi \stackrel{\text{def}}{=} \frac{\epsilon n}{4 + 2\epsilon}$. Since the Binomial distribution is symmetrical about its mean $n/2$, the inequality above is equivalent to

$$\Pr \left\{ \frac{n}{2} + r^j \geq \frac{n}{2} + \xi \right\} \leq \frac{\delta}{2},$$

According to Chernoff bound theorem, for any $X \sim \text{Binomial}(n, 1/2)$, and $0 \leq t \leq \sqrt{n}$,

$$\Pr \left\{ X \geq \frac{n}{2} + t \frac{\sqrt{n}}{2} \right\} \leq e^{-t^2/2}.$$

Rewriting $\xi = \frac{\sqrt{n}}{2} \cdot \frac{\sqrt{n\epsilon}}{2 + \epsilon}$, and replacing t with $\frac{\sqrt{n\epsilon}}{2 + \epsilon}$, X with $n/2 + r^j$, the requirement for (ϵ, δ) -DP reduces to:

$$e^{-t^2/2} \leq \frac{\delta}{2} \implies n \geq 2 \left(\frac{2 + \epsilon}{\epsilon} \right)^2 \ln \left(\frac{2}{\delta} \right). \quad (4)$$

Therefore, Theorem 1 follows. \square

It should be noted that when $\epsilon \ll 2$, $n \geq \frac{8}{\epsilon^2} \ln(\frac{2}{\delta})$, providing a constant-factor improvement over the Binomial Mechanism in [14]. Unlike Laplace or Gaussian Mechanism used in the original PATE [15], [16], Binomial Mechanism avoids floating-point representation issues and enables efficient transmission, thus it can be seamlessly used with a cryptosystem. Furthermore, the stability of Binomial distribution facilitates noise distribution among multiple teachers.

B. Discrete Gaussian Mechanism

Discrete Gaussian (DG) Mechanism belongs to the category of Gaussian Mechanism [5], hence it satisfies all the properties of Gaussian Mechanism. However, in discrete Gaussian Mechanism, discrete Gaussian noise is sampled from a discrete Gaussian by the following Definition IV.1

Definition IV.1 (Discrete Gaussian). *The pmf of the discrete Gaussian is proportional to the pdf of its continuous version. For any $x \in \mathbb{Z}$, the pmf of discrete Gaussian is defined as:*

$$P(X = x) \propto f_X(x) = \mathcal{N}(x; \mu_X, \sigma_X^2) = \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left(-\frac{(x - \mu_X)^2}{2\sigma_X^2}\right).$$

Corollary 1 (Stability of Discrete Gaussian). *The sum of independent discrete Gaussian distributed random variables still follows a discrete Gaussian distribution.*

Discrete Gaussian is a stable distribution as its continuous version (a sum of discrete Gaussian r.v.'s is still discrete Gaussian). The detailed proof for the stability of discrete Gaussian distribution can be referred to the Corollary A.1. in [17]. Like Binomial distribution, the stability of discrete Gaussian is ideal for realising DDP: analysis of overall privacy is made possible by analysing individuals, while also supporting fault tolerance if some individuals are compromised. Therefore, we utilize the stability of discrete Gaussian to distribute the noise generation among parties. To determine how much optimal noise should be added, we adopt an analytic Gaussian Mechanism as in Theorem 2, which eliminates the constraint $\epsilon < 1$ in the classical Gaussian Mechanism and removes at least a third of the variance of the noise compared to the classical Gaussian Mechanism, thus delivering better utility [18].

Theorem 2. *Let $f : \mathbb{X} \rightarrow \mathcal{R}^d$ be a function with global L_2 sensitivity Δ . For any $\epsilon \geq 0$ and $\delta \in [0, 1]$, the Analytic Gaussian Mechanism $M(x) = f(x) + Z$ with $Z \sim N(0, \sigma^2 I)$ is (ϵ, δ) -DP if and only if*

$$\Phi\left(\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) - e^\epsilon \Phi\left(-\frac{\Delta}{2\sigma} - \frac{\epsilon\sigma}{\Delta}\right) \leq \delta. \quad (5)$$

In order to obtain (ϵ, δ) -DP for a function f with global L_2 sensitivity Δ , it is enough to add Gaussian noise with variance σ^2 satisfying Eq. 5. We therefore distribute the discrete Gaussian noise at a level of σ^2 among parties.

V. DISTRIBUTED PRIVACY-PRESERVING PREDICTION

In this work, we study the applicability of DPPP to horizontally partitioned databases, where multiple parties each owns different groups of individuals with similar features. For example, different hospitals, each holding the same kind of information for different patients, can collaboratively perform statistical analyses of the union of their patients, while ensuring privacy for each patient. Consequently, hereafter, instead of training a centralized model to solve the task associated with the whole database $\mathbf{D} \in \mathbb{R}^{|D| \times d}$, $\mathbf{Y} \in \mathbb{R}^{|D|}$, the whole database \mathbf{D} is partitioned into N disjoint subsets, $\{\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_N\}$ that are held by N parties who are unwilling to make their training data, model parameters or model predictions public or share them with others. Here $|D|$ refers to the total number of training records in \mathbf{D} , \mathbf{D}_i and \mathbf{Y}_i represent party i 's training data and labels respectively. Individual models are trained separately on each subset $\{\mathbf{D}_i, \mathbf{Y}_i\}$.

In the case of prediction for any test point x , each party applies $f(x, \mathbf{D}_i) = \mathbf{y}_i$, i.e., a prediction function that returns the prediction $\mathbf{y}_i \in \{0, 1\}^c$ with the position of value 1 corresponding to the predicted class. The aggregate of multiple predictions becomes: $\sum_i f(x, \mathbf{D}_i) = \sum_i \mathbf{y}_i$, where \mathbf{y}_i is the one-hot prediction vector produced by teacher i 's local model built on individual training data \mathbf{D}_i , hence the aggregate for each class equals to the sum of N scalars. In *distributed privacy-preserving prediction* (DPPP), the goal is to privately release the aggregated prediction, i.e., the noisy sum: $\sum_i (\mathbf{y}_i + \mathbf{r}_i)$, where $\mathbf{r}_i = (r_i^1, \dots, r_i^c)$, $r_i^j = z - mp$, $z \sim B(m, p)$, $m = n/N$ and $p = 1/2$ for BM, or $r_i^j \sim DG(0, \sigma/\sqrt{N})$ where σ satisfies Eq. 5 for DGM. Our framework aims to deliver the differentially private aggregated prediction that is close to the desired aggregate $\sum_i \mathbf{y}_i$, while providing privacy guarantee.

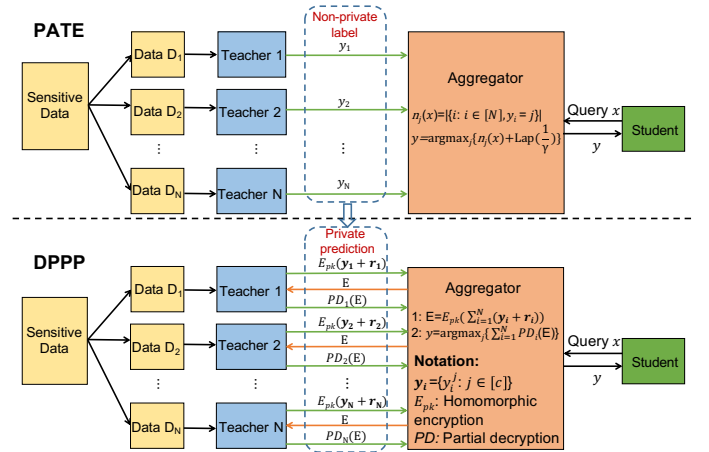


Fig. 1: Overview of the PATE and DPPP.

To realize this goal, we take inspiration from PATE framework. As illustrated in Fig. 1, both PATE and DPPP first train an ensemble of teachers on disjoint subsets of the sensitive data, then the aggregator aggregates their outputs. However, the main difference between PATE and our DPPP is that the aggregator is trusted in PATE, so teachers directly share

their non-private labels with the aggregator, while we improve PATE by eliminating the trust on the aggregator in DPPP. In particular, the total amount of noise required to guarantee (ϵ, δ) -DP of the aggregated prediction is distributed among teachers by using DDP: each teacher i adds a share of noise r_i to its local prediction y_i . The noise shares are chosen such that $\sum_{i=1}^N r_i = r$ is sufficient to ensure (ϵ, δ) -DP of the aggregated prediction, but r_i alone is not sufficient to ensure (ϵ, δ) -DP of local prediction, thus $y_i + r_i$ cannot be directly released to the aggregator. Therefore, it necessitates the help of cryptographic techniques to maintain utility and ensure aggregator obliviousness, as evidenced in [4], [11], [12]. Hence, we combine DDP with a distributed cryptosystem to achieve this goal. As shown in Fig. 1, in DPPP, each teacher i first computes the encryption of $y_i + r_i$ before sending it to the aggregator. Due to the additive homomorphic property in Eq. (2), the aggregator can compute the encryption of the noisy sum of all the local predictions as $E = E(\sum_{i=1}^N (y_i + r_i))$. This aggregated encryption E is then sent back to all teachers to compute partial decryptions. Finally, the partial decryptions $PD_i(E)$ are forwarded to the aggregator, who combines all the partial decryptions to get the final decryption, *i.e.*, the aggregated prediction.

Theorem 3. Suppose D, D' are neighboring databases that differ by one record, then each coordinate of the aggregated prediction, as given by $\sum_{i=1}^N y_i^j$, differ by at most 1. Let \mathcal{M} be the mechanism that reports $\arg \max_j \left\{ \sum_{i=1}^N (y_i^j + r_i^j) \right\}$. Then \mathcal{M} satisfies (ϵ, δ) -DP, provided $r_i^j = B(m, 1/2) - m/2, m = n/N, n \geq 2 \left(\frac{2+\epsilon}{\epsilon} \right)^2 \ln \left(\frac{2}{\delta} \right)$ or $r_i^j \sim DG(0, \sigma/\sqrt{N})$ where σ satisfies Eq. 5.

As stated in Theorem 3, the aggregated prediction is (ϵ, δ) differentially private, the privacy guarantee stems from the aggregation of teacher ensemble. If teacher i assigns class j to input x , then y_i^j equals 1 while all the other elements are all 0's. For any test point x , independent noise share r_i^j is added to each teacher's prediction for each class y_i^j . Hence, the aggregated prediction for class j is equivalent to $\sum_{i=1}^N (y_i^j + r_i^j)$, and the predicted class equals:

$$\arg \max_j \left\{ \sum_{i=1}^N (y_i^j + r_i^j) \right\}, \quad (6)$$

where $r_i^j = z - m/2, z \sim B(m, 1/2), m = n/N$, here n is the total number of tosses in Theorem 1 for BM, or $r_i^j \sim DG(0, \sigma/\sqrt{N})$ where σ satisfies Eq. 5 for DGM. When there is a strong consensus among N teachers, the label they almost all agree on (maximum of the aggregated prediction) does not depend on any particular teacher. Overall, DPPP provides a differentially private API: the privacy cost of each aggregated prediction made by the teacher ensemble is known. Semi-supervised learning can be further used to train a student model given a limited set of labels from the aggregation mechanisms [15], [16].

VI. DISTRIBUTED CRYPTOSYSTEM

As part of DPPP, based on the threshold Paillier cryptosystem [19], we design a secure aggregation protocol in Protocol 1, which can calculate the summation of teachers' local predictions without disclosing any of them. As we can see, the protocol mainly executes in two phases. In the first phase, the aggregator aggregates the encrypted noisy predictions as $E_{pk}(\sum_{i=1}^N \hat{y}_i)$. Then in the second phase, a distributed decryption process is run to recover the aggregated noisy predictions $\sum_{i=1}^N \hat{y}_i$. In this protocol, what the aggregator received from all teachers are the encrypted noisy predictions and partial decryptions. Moreover, all the calculations on the aggregator are conducted on the encrypted data. What the aggregator can know is only the summation of all teachers' noisy predictions, based on which each teacher's local prediction y_i cannot be inferred, thereby providing aggregator obliviousness and significantly reducing privacy leakage. Note that the key generation needs to be done only once, hence secret-sharing protocols can be used for this purpose.

Protocol 1 Secure aggregation protocol

- 1) Each teacher i encrypts its noisy prediction \hat{y}_i as $E_{pk}(\hat{y}_i)$ as per Eq. (1), and sends it to the aggregator;
 - 2) The aggregator computes $c = E_{pk}(\sum_{i=1}^N \hat{y}_i) = \prod_{i=1}^N E_{pk}(\hat{y}_i)$ based on Eq. (2);
 - 3) The aggregator sends c to the randomly chosen t teachers;
 - 4) Each selected teacher i calculates a partial decryption based on Eq. (3), and sends it to the aggregator;
 - 5) The aggregator combines all the partial decryptions to get the summation $\sum_{i=1}^N \hat{y}_i$.
-

Fault Tolerance and Collusion. In real system, one or more teachers might fail to respond or drop out the system at some point before the completion of the protocol for several different reasons. We also consider the threat of collusion among teachers, including the aggregator, through the trust parameter t – the minimum number of honest teachers. Our proposed DPPP can be made robust to the fault tolerance and collusion of less than 1/3 compromised teachers by adopting the following two solutions: (i) (N, t) -threshold decryption [10] requires the cooperation of at least $t = \gamma N$ honest teachers for decryption, where γ is the fraction of uncompromised teachers in Definition 1. If f teachers fail to send their partial decryptions, (N, t) -threshold decryption ensures that a decryption can still be computed as long as $f < N - t$; (ii) during the noise addition, for BM, each honest teacher sets its binomial noise as $z - m/2$, where $z \sim B(m, 1/2), m = 3n/2N$; for DGM, $r_i^j \sim DG(0, \sigma/\sqrt{2N/3})$ where σ satisfies Eq. 5, *i.e.*, leaving out 1/3 teachers' randomness is still sufficient to ensure differential privacy. We remark that the number of honest parties t could be set as per different applications, and (N, t) -threshold Paillier cryptosystem requires $2 \leq t \leq N$. Here the assumption of less than 1/3 compromised parties is often practical enough in most real scenarios.

VII. PERFORMANCE EVALUATION

Comparison frameworks. To demonstrate the effectiveness of our DPPP, we compare it with the following frameworks: (1) *Centralized non-private framework* requires all teachers to pool their training data into the aggregator to train a global model; (2) *Distributed non-private framework* excludes both DP and cryptosystem, teachers directly share their local predictions with the aggregator; (3) *Local differentially private (LDP) framework* excludes cryptosystem but requires each teacher to add the required level of noise to ensure (ϵ, δ) -LDP. The added noise is of the same level as the aggregated noise in DPPP, hence much more noise is added compared to the noise added in DPPP; (4) *Standalone framework* allows teachers to individually train local models without any collaboration, and an end user directly sends a test query to one teacher, then local prediction is released under the guarantee of (ϵ, δ) -LDP; (5) PATE which relies on a trusted aggregator to add Laplace noise to the aggregated vote counts [15].

Datasets. For fair comparison with PATE, we first adopt MNIST dataset, which consists of 60K training examples and 10K testing examples¹. We also investigate the other two real-world datasets. One is Breast Cancer dataset² that contains total 569 records with 32 features, each record is classified into two classes: malignant and benign. We randomly sampled 2/3 examples from the whole database as the training set, while the remaining 1/3 as the test set. The other one is NSL-KDD dataset used for intrusion detection³, which contains total 125973 records with 41 features, each record is classified into two classes: anomaly and normal. We select a smaller subset called NSL-KDD-20 (with 20% train and test data sampled from NSL-KDD).

Experimental Setup. For MNIST, we follow [15] to use a convolution NN (CNN) model. Each teacher trains a convolutional network with two convolutional layers and one fully connected layer. For the other datasets, we use SVM model with RBF kernel. To simulate the situation in which each teacher constitutes only a limited subset of the whole database, all the training records are randomly distributed among multiple teachers such that each teacher receives nearly the same amount of records. Following the rationales provided in [15], we *empirically find appropriate values of N* for all the datasets by measuring the test accuracy of each teacher trained on one of the N partitions of the whole training set, *i.e.*, we trained ensembles of 250, 100, 100 and 20 teachers for the MNIST, NSL-KDD-20 and Breast Cancer datasets respectively. How the number of teachers affects privacy cost can be referred to [15]. We run each experiment for 20 times and report the average result. γ is set to be 1 or 2/3, *i.e.*, assuming no compromised teachers, or at most 1/3 compromised teachers. Since δ in BM and DGM are different

from the classical DP mechanisms, we choose small values from $1e^{\{-5, -4, -3, -2\}}$.

Experimental Results. We first fix $\delta = 1e^{-3}$ and report the prediction accuracy of student queries under varying privacy budgets $\epsilon \in [0.01, 1]$ in Fig. 2. In particular, the result of the standalone framework is derived by averaging over all teachers. As evidenced by Fig. 2, DPPP outperforms both the standalone and LDP frameworks, for all datasets. The centralized and distributed non-private frameworks achieve similar accuracy, indicating that the distributed non-private framework incurs minor accuracy degradation compared with the centralized non-private framework. Moreover, DPPP yields comparable accuracy to the distributed non-private framework when $\epsilon \geq 0.05$ for MNIST and NSL-KDD-20 datasets, which is also comparable to PATE where each query has a low privacy budget of $\epsilon = 0.05$ [15]. We also notice that compared with other datasets, Breast Cancer dataset requires a higher value of ϵ to achieve comparable accuracy. One reason is the limited available data split among smaller number of teachers, while compensating for the introduced noise in Eq. (6) requires large ensembles. We also observe that DPPP with DGM consistently outperforms DPPP with BM, and achieves comparable performance to the PATE. The extra noise introduced to guarantee privacy under 1/3 compromised teachers indeed slightly degrades accuracy, especially when ϵ is small. These findings provide empirical supports to guide the deployment of DPPP framework.

We further show how the values of δ impact the performance by varying $\delta \in 1e^{\{-5, -4, -3, -2\}}$. We choose MNIST and Breast Cancer for illustration, and adopt the fixed $\epsilon = 0.05$ for MNIST and $\epsilon = 0.5$ for Breast Cancer. As can be observed in Fig. 3, the effectiveness of our proposed DPPP persists, and for the fixed ϵ , varying δ follows the similar trend as varying ϵ . However, there is less difference with different δ , which agrees with the findings reported in [20]. In contrast, ϵ has larger impact on accuracy.

Computation and Communication Overhead. We use the typical 1024-bit key size and implement a $(N, \lfloor \frac{2}{3}N \rfloor)$ -threshold Paillier cryptosystem using Paillier Encryption Toolbox⁴. The average computation time at a teacher is independent of the number of teachers and remains nearly constant. On the other hand, the time required by the aggregator might increase with the number of teachers, but this can be reduced by running the aggregator and teachers in parallel through MapReduce. In all cases, the computation overhead is quite small (within \sim ms), most of which is spent on cryptographic operations. For communication complexity, a Paillier ciphertext is estimated as 2048 bits (256 bytes). Therefore, the total communication cost between the aggregator and any teacher can be estimated as $256 \times c \times 3 = 768 \times c$ bytes, where c is the number of classes and 3 refers to three rounds of communication in Fig. 1, which is also well within the realm of practicality as c is usually small.

¹<http://yann.lecun.com/exdb/mnist/>

²[https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+\(Diagnostic\)](https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic))

³<https://www.unb.ca/cic/datasets/nsf.html>

⁴<http://cs.utdallas.edu/dspl/cgi-bin/pailliertoolbox/>

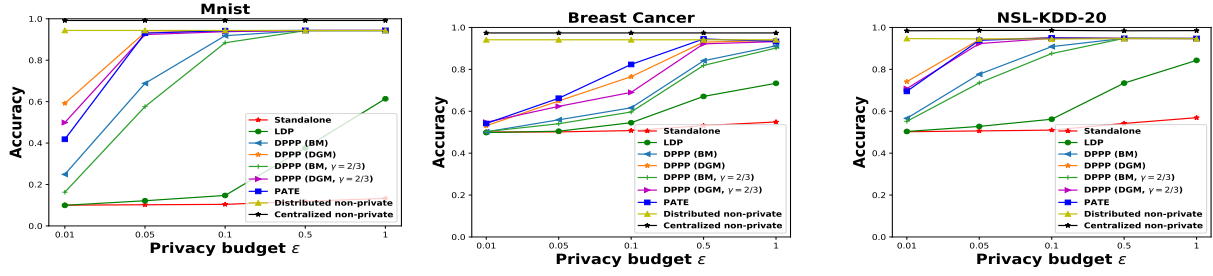


Fig. 2: Prediction accuracy of student queries for all datasets with varying ϵ .

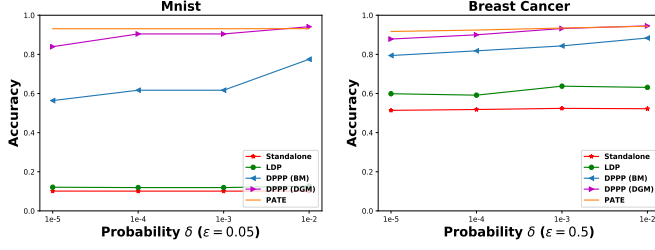


Fig. 3: Prediction accuracy of student queries with varying δ .

Discussion. Different from the typical balanced and IID data distribution, in real practice, due to the differences in sensor quality, ambient noise, and skill level, the collected data by each teacher might be: (1) Unbalanced: due to the capabilities of different teachers, some teachers may have large training data, while others have little or no data. For substantially unbalanced data, most teachers have only a few examples, and a few teachers have a large number of examples. (2) Non-IID: the collected data by each teacher might not be representative of the population distribution.

These two aspects are usually considered in federated learning, and might affect the accuracy of DPPP, especially when most teachers have few or extremely unrepresentative examples. To improve robustness to unbalanced and/or non-IID data distributions, current methods allow teachers to share locally trained model updates with the aggregator [21], [22], but giving the aggregator access to all teachers’ updates clearly risks privacy leakage. To privately share individual model updates, Bonawitz *et al.* [23] proposed a secure aggregation protocol to securely aggregate local model updates as the weighted average to update the global model on the aggregator. However, this incurs both extra computation and communication costs.

VIII. CONCLUSION AND FUTURE WORK

We have presented a distributed privacy-preserving prediction framework, which enables multiple parties to collaboratively deliver more accurate predictions through an aggregation mechanism. Distributed differential privacy via Binomial Mechanism or Discrete Gaussian Mechanism and homomorphic encryption are combined to preserve individual

privacy, maintain utility and ensure aggregator obliviousness. For the Binomial Mechanism, we offer tighter bounds than that in the previous works. Preliminary analysis and performance evaluation confirm the effectiveness of our framework. We plan to extend our framework to the unbalanced and non-IID data distribution. We also expect to extend our framework to various machine learning scenarios beyond classification. It is also important to investigate how to conduct privacy accounting for many subsequent queries by using different DDP mechanisms.

REFERENCES

- [1] Lucila Ohno-Machado, Paulo Sérgio Panse Silveira, and Staal Vinterbo, “Protecting patient privacy by quantifiable control of disclosures in disseminated databases,” *International journal of medical informatics*, vol. 73, no. 7, pp. 599–606, 2004.
- [2] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, “Membership inference attacks against machine learning models,” in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 3–18.
- [3] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim, “Practical privacy: the SuLQ framework,” in *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2005, pp. 128–138.
- [4] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, and Dawn Song, “Privacy-preserving aggregation of time-series data,” in *NDSS*. Internet Society., 2011.
- [5] Cynthia Dwork, Aaron Roth, et al., “The algorithmic foundations of differential privacy,” *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.
- [6] Lingjuan Lyu, Yee Wei Law, Jiong Jin, and Marimuthu Palaniswami, “Privacy-preserving aggregation of smart metering via transformation and encryption,” in *2017 IEEE Trustcom*. IEEE, 2017, pp. 472–479.
- [7] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam, “Local differential privacy and its applications: A comprehensive survey,” *arXiv preprint arXiv:2008.03686*, 2020.
- [8] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta, “Amplification by shuffling: From local to central differential privacy via anonymity,” in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2019, pp. 2468–2479.
- [9] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim, “The privacy blanket of the shuffle model,” in *Annual International Cryptology Conference*. Springer, 2019, pp. 638–667.
- [10] Ivan Damgård and Mads Jurik, “A generalisation, a simplification and some applications of Paillier’s probabilistic public-key system,” in *International Workshop on Public Key Cryptography*. Springer, 2001, pp. 119–136.
- [11] Vibhor Rastogi and Suman Nath, “Differentially private aggregation of distributed time-series with transformation and encryption,” in *SIGMOD*. ACM, 2010, pp. 735–746.

- [12] Gergely Ács and Claude Castelluccia, “I have a dream!(differentially private smart metering).,” in *Information hiding*. Springer, 2011, vol. 6958, pp. 118–132.
- [13] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan, “cpsgd: Communication-efficient and differentially-private distributed sgd,” in *Advances in Neural Information Processing Systems*, 2018, pp. 7564–7575.
- [14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor, “Our data, ourselves: Privacy via distributed noise generation,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 486–503.
- [15] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar, “Semi-supervised knowledge transfer for deep learning from private training data,” in *ICLR*, 2017.
- [16] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson, “Scalable private learning with pate,” in *ICLR*, 2018.
- [17] Lingjuan Lyu, *Privacy-preserving machine learning and data aggregation for Internet of Things*, Ph.D. thesis, The University of Melbourne, 2018.
- [18] Borja Balle and Yu-Xiang Wang, “Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising,” *arXiv preprint arXiv:1805.06530*, 2018.
- [19] Ronald Cramer, Ivan Damgård, and Jesper B Nielsen, “Multiparty computation from threshold homomorphic encryption,” in *International conference on the theory and applications of cryptographic techniques*. Springer, 2001, pp. 280–300.
- [20] Martín Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, “Deep learning with differential privacy,” in *CCS*. ACM, 2016, pp. 308–318.
- [21] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Aguera y Arcas, “Federated learning of deep networks using model averaging,” *arXiv preprint arXiv:1602.05629*, 2016.
- [22] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang, “Learning differentially private recurrent language models,” in *ICLR*, 2018.
- [23] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth, “Practical secure aggregation for privacy-preserving machine learning,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.