Distributed Privacy-Preserving Prediction

Lingjuan Lyu, Yee Wei Law, Kee Siong Ng

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 secure privacy guarantee. Our framework integrates two main techniques to guarantee individual privacy. First, we improve the previous analysis of the Binomial mechanism to achieve distributed differential privacy. Second, we utilize homomorphic encryption to ensure that the aggregator learns nothing but the noisy aggregated prediction. We empirically evaluate the effectiveness of our framework on various datasets, and compare it with other baselines. The 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—Distributed framework, Privacy-preserving, Prediction.

1 INTRODUCTION

In real world, many practical applications would benefit from large-scale machine learning across sensitive datasets owned by different parties. This trend is motivated by the fact that the data owned by a single organization may be very homogeneous, 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. A common practice for distributed learning is to parallelize computation among multiple parties to help reduce the demand for resources on any single party, while reaping benefits from multiple data resources.

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 model 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.

In consideration of privacy concerns in a distributed setting, instead of sharing more sensitive local data or model parameters of any party, we examine an alternative approach which ensures that the aggregated prediction is differentially private, without compromising privacy or degrading test accuracy. As such, we are motivated to build a distributed privacy-preserving prediction (DPPP) framework, which distinguishes itself from existing techniques by allowing parties to keep full control of their own data, and distributes the total amount of noise required to guarantee global differential privacy (GDP) of the final prediction among multiple parties.

In particular, 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 $D \in \mathbb{R}^{r \times d}$, $Y \in \mathbb{R}^r$, the whole database $D$ is partitioned into $N$ disjoint subsets, $\{D_1, D_2, ... D_N\}$ 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 $r$ refers to the total number of training records in $D$, $D_i$ and $Y_i$ represent party $i$’s training data and labels respectively. Individual models are trained separately on each subset $\{D_i, Y_i\}$.

To better preserve privacy, we investigate a predicate function $f$ that releases model prediction for any test record. In the distributed scenario, the answer to each test record is obtained by applying $f$ to the disjoint subsets and summing their predictions. Considering individual privacy and potential black-box attacks [4], we appropriately perturb model predictions before releasing them for aggregation. The primitive of this provably private noisy sum is demonstrated by Blum et al. [5]. Each party first trains a local model based on its training data $D_i$, to answer the prediction query for any test point $x$, each party applies a predicate function $f$ to $x$, which returns the votes for all $c$ classes, i.e., $f(x, D_i) = y_i$, where $y_i \in \{0,1\}^c$ is an one-hot prediction vector that sums up to 1. Hence, the aggregated prediction for each class falls within the range $\{0, N\}$, where $N$ is total number of parties. It then follows that the sensitivity of the aggregate is 1 with respect to the change of any record, corresponding to the scenario when exactly one party changes its data. In other words, changing one record can at most affect one party’s prediction, thus the aggregated prediction for each class can be changed by at most 1. To further minimize the privacy leakage.
from local prediction $f(x, D_i)$, maintain utility and ensure aggregator obliviousness, we combine distributed differential privacy and homomorphic encryption. This simple privacy-preserving aggregation of local model predictions is agnostic to the underlying machine-learning techniques (cf. differentially-private stochastic gradient descent in [6]).

Our contributions are summarized as follows:

- 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) to guarantee $(\epsilon, \delta)$-differential privacy in the distributed setting, and formally provide a tightest bound to date.
- The experimental results on various datasets demonstrate that DPPP delivers comparable performance to the non-private frameworks, and yields better results than the local differentially private and standalone frameworks.

## 2 PRELIMINARIES AND RELATED WORK

### 2.1 Distributed Differential Privacy

Given that the aggregator evaluates a function $h : \mathcal{D}^n \rightarrow \mathcal{R}$ on randomized result $\hat{y}_i \in \mathcal{D}^n$ of $n$ parties (here we use hatted variables to denote the randomized versions of a party’s local result), and a set of compromised parties $K$, we let $r_K := \{ r_i : i \in K \}$ and denote $\overline{K}$ be the complement of $K$, i.e., uncompromised set $\overline{K} = \{ 1, 2, ..., N \} \setminus K$. A formal DDP definition adapted from [7] is given as follows.

**Definition 1** ($(\epsilon, \delta, \gamma)$-Distributed DP). Let $\epsilon > 0$, $0 \leq \delta < 1$ and $0 < \gamma < 1$. We say that the data randomization procedure with randomness over the joint distribution of $r := (r_1, \ldots, r_N)$ preserves $(\epsilon, \delta, \gamma)$-differential privacy (DDP), with respect to the function $h$ and under $\gamma$ fraction of uncompromised parties 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 $K$ of at least $\gamma n$ honest parties,

$$\Pr(\mathcal{M}(D) \in S | r_K) \leq \exp(\epsilon) \cdot \Pr(\mathcal{M}(D') \in S | r_K) + \delta.$$  

In the above definition, the probability is conditioned on the randomness $r_K$ from compromised parties, i.e., it ensures that if at least $\gamma N$ participants are honest and uncompromised, we will accumulate noise of a similar magnitude as that of the GDP. For differentially-privately aggregate of local statistics, DDP permits each party to randomise its local statistic to a lesser degree than would local differential privacy (LDP).

### 2.2 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 [3] 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

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

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 homomorphic properties of this cryptosystem can be described as:

$$E_{pk}(m_1 + m_2) = E_{pk}(m_1) \cdot E_{pk}(m_2) = g^{m_1 + m_2} \cdot r_{1,2} \mod n^2,$$

$$E_{pk}(a \cdot m_1) = E_{pk}(m_1)^a = g^{am_1} \cdot r_1^n \mod n^2,$$

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

In this paper, $(N, t)$-threshold Paillier cryptosystem is adopted, in which the private key $sk$ is distributed among $N$ parties (denoted as $\{sk_1, sk_2, \cdots, sk_N\}$), 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:

$$c_i = c^{2N \cdot sk_i}$$

Then based on the combining algorithm in [8], at least $t$ partial decryptions are required to recover the plaintext $m$.

### 2.3 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 the central server to ensure LDP. 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 by the following schemes:

Rastogi et al. [9] allow an untrusted aggregator to privately sum over multiple data sources by combining distributed differential privacy and Paillier encryption. However, their proposed Distributed Laplace Perturbation Algorithm (DLPA) generates Laplace noise using four Gaussian variables, while our solution is more efficient and simpler by exploiting the stability property of Binomial distribution.

Acs et al. [10] propose to differential-privately aggregate smart metering data over multiple slots. Smart meters are grouped into clusters, where each cluster covers thousands of smart meters corresponding to a quarter of a city. The Laplace noise-tainted readings are sent to an electricity distributor, but the requirement that all meters in a cluster share pairwise keys largely limits its applicability.

Shi et al. [7] formalize the notion of distributed differential privacy (DDP) to ensure $(\epsilon, \delta)$-differential privacy of the aggregate of time-series data in every time period, where noise is sourced from multiple participants. The symmetric
geometric distribution is used as a discrete approximation to the Laplace distribution, where each participant adds noise probabilistically. However, their encryption scheme relies on a trusted dealer allocating \( g + 1 \) secrets that sum to 0, to the aggregator and \( g \) participants. Moreover, their construction is not robust against malicious nodes and node failures.

More recently, Agarwal et al. [11] demonstrate that Binomial mechanism can achieve nearly the same utility as the Gaussian mechanism. 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 didn’t provide a complete scheme to protect against the untrusted aggregator.

In the absence of an aggregator, a distributed implementation is highly desired. The most relevant one called “Our Data, Ourselves” (ODO) [12] is used to generate shares of random noise. For example, the shares of random Binomial noise can be generated by coin flipping, which is secure against malicious parties; however, it requires communication among parties and the expensive verifiable secret sharing technique results in \( O(N) \) multiplications and additions in shares, where \( N \) is the number of parties, thus not preferable for a large number of parties.

Another line of work is Private Aggregation of Teacher Ensembles (PATE) proposed by Papernot et al. [13]. 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 an important issue in their design, i.e., the protection of individual teachers’ private information. In practical scenarios like federated learning [14], [15], these teachers could be mobile participants with high privacy standards.

3 PROBLEM DEFINITION

Similar as Shi et al. [7], 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’s data 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 work [7], [9], [10], to ensure the correctness and functionality of the system, we do not consider protocol breakers, as protocol breakers may not be desirable in many practical settings, and are not in the commercial interest of collaborative service providers to provide prediction service. We remark that our privacy model is stronger than [7] in the sense that we do not trust the aggregator, and our framework is also robust to party failures, as indicated in Section 6. This assumption is often more realistic, as hospitals or financial institutions are probably not willing to disclose their private statistics to any third party. And one or more parties may fail to upload their encrypted values or fail to respond in the form of a Denial-of-Service attack.

In particular, 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. Similar to PATE, we first train an ensemble of teachers on disjoint datasets, then a teacher aggregator aggregates local predictions instead of local data or local model parameters, which makes our approach applicable to any arbitrary and mixed classifiers, i.e., the classifiers of teachers could be different. To preserve privacy and maintain utility, we combine distributed differential privacy (DDP) and homomorphic encryption. To be specific, for DDP, we distribute the noise generation task among multiple teachers, who jointly contribute randomness to ensure differential privacy of the global statistic, i.e., aggregated prediction. While a teacher may add less noise, the aggregated noise must meet the required level to guarantee the privacy of the global statistic upon release. Hence, the ultimate goal is to release a differentially-private global statistic, i.e., noisy aggregated prediction \( \sum_i f(x, D_i) + r \), where \( r = \sum_i r_i \) is the required noise to guarantee \((\epsilon, \delta)\)-differential privacy of the aggregated prediction. Moreover, since cryptographic protocol requires discrete inputs, instead of adding floating-point Gaussian noise to each individual’s prediction, we leverage a Binomial mechanism to generate discrete Binomial noise, as described in the next section.

4 BINOMIAL MECHANISM

The Binomial Mechanism 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 the Binomial Mechanism (BM) for the query \( f \) with the output space in \( \{0, 1\}^c \), where \( c \) is the number of classes, i.e., for each data point, a predicate function \( f \) 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 \( D_i \) and predicate \( f \): let \( f(x, D_i) = y_i \) be the local prediction vector produced by party \( i \) given data \( D_i \), and \( y_i^j \) be the \( j \)-th element of \( y_i \), i.e., prediction (vote status) for class \( j \). If party \( i \) assigns class \( j \) to input \( x \), then \( y_i^j = 1 \) while other elements are all 0’s, i.e., \( y_i^l = 0 \) for \( l \in \{1, 2, ..., c\} \setminus \{j\} \). The noisy vote count \( \tau \) on each class \( j \) equals \( \tau = \sum_{i=1}^{n} y_i^j + \text{noise} \), replacing the whole database \( D \) with \( D' \) differing only in one row changes the summation in each class by at most 1. Bounding the ratio of probabilities that \( \tau \) occurs with inputs \( D \) and \( D' \) amounts to bounding the ratio of probabilities that \( \text{noise} = r^j \) and \( \text{noise} = r^j + 1 \), for different possible ranges of values of \( r^j \). Given predicate \( f(x, D_i) = y_i \), 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 \((\epsilon, \delta)\)-differentially private so long as the total number of losses \( n \geq 2 \left( \frac{2\ln 2}{\epsilon^2} \right)^2 \ln \left( \frac{2}{\delta} \right) \). Note that this lower bound is tighter than \( n \geq 64 \ln \left( \frac{2}{\delta} \right) / \epsilon^2 \) given in [12], 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 \( r' \) is the random Binomial noise added to the vote count for each class \( j \), then
\[
\tau = \sum_{i=1}^{N} y'_i + r; \quad r' = \sum_{i=1}^{N} y'_i + r + 1.
\]

For the Binomial random variable with bias \( 1/2 \), whose mass at \( \frac{n}{2} + r' \) is
\[
\Pr\left(\frac{n}{2} + r\right) = \left(\frac{n}{2} + r\right) \left(\frac{1}{2}\right)^n,
\]
\( \epsilon \)-differential privacy requires that
\[
\Pr\left\{\frac{n}{2} + r\right\} \leq e^\epsilon \Pr\left\{\frac{n}{2} + r + 1\right\}
\]
\[
\Rightarrow \left(\frac{n}{2} + r\right) \left(\frac{1}{2}\right)^n \leq e^\epsilon \left(\frac{n}{2} + r + 1\right) \left(\frac{1}{2}\right)^n
\]
\[
\Rightarrow \frac{n}{2} + r + 1 \leq e^\epsilon \left(\frac{n}{2} - r\right).
\]

To express \( r' \) in terms of \( \epsilon \) in an algebraically simple way, we use the inequality \( 1 + \epsilon \leq e^\epsilon \):
\[
\frac{n}{2} + r + 1 \leq (1 + \epsilon) \left(\frac{n}{2} - r\right) \Rightarrow r' \leq \frac{en - 2}{4 + 2\epsilon} < \frac{en}{4 + 2\epsilon}.
\]

Therefore, the Binomial random variable \( n/2 + r' \) can apparently achieve \( \epsilon \)-differential privacy, as long as \( r' < \frac{en}{4 + 2\epsilon} \).

Note:
- \( n/2 + \frac{en}{4 + 2\epsilon} = \left(1 + \frac{\epsilon}{2+\epsilon}\right) \frac{n}{2} \). This equation will be used in Eq. (5) later.
- This noise upper bound is tighter than Dwork et al.’s [12].

However, note that when \( r' \) exceeds this upper bound, \( \epsilon \)-differential privacy will be violated. Hence, we turn to the relaxed \((\epsilon, \delta)\)-differential privacy, which requires that
\[
\Pr\left\{\frac{n}{2} + r' \leq \frac{n}{2} - \xi \text{ or } \frac{n}{2} + r' \geq \frac{n}{2} + \xi\right\} \leq \delta,
\]
where \( \xi \overset{\text{def}}{=} \frac{en}{4 + 2\epsilon} \). Since the Binomial distribution is symmetrical about its mean \( n/2 \), the inequality above is equivalent to
\[
\Pr\left\{\frac{n}{2} + r' \geq \frac{n}{2} + \xi\right\} \leq \delta/2,
\]
According to Chernoff bound theorem, for any \( X \sim \text{Binomial}(n, 1/2) \), and \( 0 \leq t \leq \sqrt{n} \),
\[
\Pr\left\{X \geq \frac{n}{2} + t\sqrt{\frac{n}{2}}\right\} \leq e^{-t^2/2}.
\]
Rewriting \( \xi = \frac{\sqrt{2\epsilon}}{2} \cdot \frac{\sqrt{n}}{\sqrt{2\epsilon}} \), and replacing \( t \) with \( \frac{\sqrt{n}}{\sqrt{2\epsilon}} \), \( X \) with \( n/2 + r' \), the requirement for \((\epsilon, \delta)\)-differential privacy reduces to:
\[
e^{-t^2/2} \leq \frac{\delta}{2} \Rightarrow n \geq 2 \left(\frac{2 + \epsilon^2}{\epsilon}\right) \ln\left(\frac{2}{\delta}\right).
\]
Therefore, Theorem 1 follows.

It should be noted that when \( \epsilon \ll 2, n \geq \frac{8}{\epsilon^2} \ln\left(\frac{2}{\delta}\right) \), which shows a constant-factor improvement over the original Binomial mechanism given in [12]. It also implies that the Binomial and Gaussian Mechanism perform identically as \( \epsilon \to 0 \).

Unlike Laplace or Gaussian distribution used in original PATE [13], [16], Binomial distribution avoids floating-point representation issues and enables efficient transmission, thus it can be seamlessly used with cryptosystem. Furthermore, the stability of Binomial distribution, as stated in Lemma 1 facilitates the noise distribution among multiple teachers.

Lemma 1. (Stability of Binomial distribution) If \( X \sim B(n, p) \) and \( Y \sim B(m, p) \) are independent Binomial variables with the same probability \( p \), then \( X + Y \) is also a Binomial variable, with distribution following \( Z = X + Y \sim B(n + m, p) \).

5 DISTRIBUTED PRIVACY-PRESERVING PREDICTION

In the case of prediction for any unlabeled public test point \( x \), \( f(x, D_i) = y_i, i.e., a predicate function that given a test record \( x \) returns the prediction \( y_i \in \{0, 1\} \). The aggregate of multiple predictions becomes: \( \sum_i f(x, D_i) = \sum_i y_i \), where \( y_i \) is the one-hot prediction vector produced by teacher \( i \)'s local model built on individual training data \( D_i \), 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., noisy sum: \( \sum_i f(x, D_i) + r_i \), where \( r_i = (r_{i1}, \cdots, r_{in}) \), \( r_{ij} \sim \mathcal{N}(0, \sigma^2) \), \( \sigma \sim B(m, p), m = n/N, p = 1/2 \). Our framework aims to deliver the differentially private aggregated prediction that is close to the desired aggregate \( \sum_i f(x, D_i) \), while providing privacy guarantee.

To realize this goal, we take inspiration from PATE framework, which first trains an ensemble of teachers on disjoint subsets of the sensitive data, then the aggregator aggregates their outputs, this makes our approach applicable to any arbitrary and mixed classifiers. Moreover, we minimize the need for parties to trust any aggregator, hence improving PATE as follows: teachers add shares of noise to their predictions before forwarding to the aggregator. Aggregation at the aggregator results in an aggregated prediction with the noise accumulated from all teachers. Hence, each teacher may add less noise, as long as the aggregated noise in the aggregated prediction meets the required noise level, the privacy of the final output (aggregated prediction) can be guaranteed, as stated in Theorem 2 which follows immediately from Theorem 1 and Lemma 1.

Theorem 2. Suppose \( D, D' \) are neighboring databases that differ by one record, then each coordinate of the aggregated prediction, as given by \( \sum_{i=1}^{N} y'_i \), differ by at most 1. Let \( M \) be the mechanism that reports \( \arg \max_j \left\{ \sum_{i=1}^{N} (y'_i + r'_i) \right\} \). Then \( M \) satisfies \((\epsilon, \delta)\)-differential privacy, provided \( r'_i \sim B(m, 1/2) - m/2, m = n/N, \delta \geq \left(\frac{1+\epsilon}{\epsilon}\right)^2 \ln\left(\frac{2}{\delta}\right) \).

The privacy guarantee stems from the aggregation of teacher ensemble. If teacher \( i \) assigns class \( j \) to input \( x \), then \( y'_i \) equals 1 while all the other entries are all 0’s. For any
test point $x$, independent noise share $r^j_i$ is added to each teacher’s prediction for each class $y^j_i$. Hence, the aggregated prediction for class $j$ is equivalent to $\sum_{i=1}^{N} (y^j_i + r^j_i)$, and the aggregator outputs the predicted class as:

$$\arg\max_j \left\{ \sum_{i=1}^{N} (y^j_i + r^j_i) \right\}, \quad (6)$$

where $r^j_i = z - m/2$, $z \sim B(m, 1/2)$, $m = n/N$, and $n$ is the total number of tosses in Theorem 1. 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 model. 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 used to train a student model with comparable utility given a limited set of labels from the aggregation mechanism [13], [16].

The frameworks of PATE and DPPP are illustrated in Fig. 1. As can be observed, the main difference between the original PATE and our DPPP is that teachers directly share non-private labels in PATE, while in DPPP, each teacher releases an one-hot prediction vector $y^j_i \in \{0, 1\}^c$ with the position of value 1 corresponding to the predicted class. For example, for anomaly detection with two classes, each teacher outputs prediction $y^j_i \in \{0, 1\}^2$ for any test point $x$, indicating whether $x$ is an anomaly or not.

To preserve privacy of the final output (aggregated prediction) before releasing it to the student, we distribute the total amount of noise using DDP: each teacher $i$ adds a share of noise $r^j_i$ to its local prediction $y^j_i$. The noise shares are chosen such that $\sum_{i=1}^{N} r^j_i = r$ is sufficient to ensure $(\epsilon, \delta)$-differential privacy of the aggregated prediction, but $r^j_i$ alone is not sufficient to ensure $(\epsilon, \delta)$-differential privacy of local prediction, thus $y^j_i + r^j_i$ cannot be directly released to the aggregator. Therefore, this necessitates the help of cryptographic techniques to maintain utility and ensure aggregator obliviousness, as evidenced in [7], [9], [10], [17].

In particular, we combine DDP with a distributed cryptosystem to achieve this goal. As shown in Fig. 1 for DPPP, each teacher $i$ first computes the encryption of $y^j_i + r^j_i$ before sending it to the aggregator. The aggregator computes the encryption of the noisy sum of all the local predictions as $E = E(\sum_{i=1}^{N} (y^j_i + r^j_i))$. This aggregated encryption $E$ is then sent back to all teachers who then use the threshold property to compute their respective decryption shares. Finally, the decryption shares $PD_i(E)$ are forwarded to the aggregator, who combines all decryption shares to get the final decryption, i.e., the aggregated prediction. Due to homomorphic encryption, DPPP provides aggregator obliviousness, i.e., the aggregator cannot learn anything about $y^j_i$, the only information the aggregator can learn is the aggregated noisy prediction, thereby significantly reducing privacy leakage.

6 DISTRIBUTED CRYPTO SYSTEM

As part of DPPP, we design a secure aggregation protocol based on the threshold Paillier cryptosystem [18]. As shown in Protocol 1, the proposed secure aggregation protocol can calculate the summation of teachers’ local predictions without disclosing any of them.

![Diagram](image-url)

**Fig. 1.** Overview of the PATE and DPPP: (1) an ensemble of teachers is trained on disjoint subsets of the sensitive data, (2) the query from the student is labeled using the ensemble.

**Protocol 1 Secure aggregation protocol**

| Input | Noisy prediction from each teacher $\hat{y}_i$. |
|-------|-----------------------------------------------|
| Output| The aggregated noisy prediction $\sum_{i=1}^{N} \hat{y}_i$ |

1) Each teacher $i$ encrypts its noisy prediction according to Eq. (1), and sends the ciphertext $E_{pk}(\hat{y}_i)$ to the aggregator;  
2) The aggregator computes $c = \prod_{i=1}^{N} E_{pk}(\sum_{i=1}^{N} \hat{y}_i)$ 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. (4), and sends it to the aggregator;  
5) The aggregator combines all the partial decryptions from the chosen $t$ teachers to get the summation $\sum_{i=1}^{N} \hat{y}_i$.

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 cannot be inferred. Note that the key generation needs to be done only once, hence secret-sharing protocols can be used for this purpose.

Our proposed DPPP can be made robust to the fault tolerance of less than $1/3$ compromised teachers by adopting the following two solutions: (i) $(N, t)$-threshold decryption [8], [19] requires the cooperation of at least $t$ honest teachers for decryption, where $\gamma$ is the fraction of uncompromised teachers in Definition 1. If $f$ teachers fail to send their decryption shares, $(N, t)$-threshold decryption ensures that a decryption can still be computed as long as $f < N - t$. Threshold Paillier cryptosystem can tolerate...
the passive corruption of \( f < N/3 \) teachers; (ii) during the noise addition, each teacher sets its binomial noise as \( z - m/2 \), where \( z \sim B(m, 1/2) \), \( m = 3n/2N \), 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.

7 Performance Evaluation

To demonstrate the effectiveness of our proposed DPPP, we compare it with the following four baseline frameworks.

Centralized non-private framework requires all teachers to send their local training data to the aggregator to train a global model. This framework delivers maximum utility, but minimum privacy.

Distributed non-private framework excludes both DP and cryptosystem, teachers directly share their local predictions with the aggregator.

Local differentially private (LDP) framework excludes cryptosystem but requires each teacher to add the required level of noise to ensure \((\epsilon, \delta)\)-LDP. The added noise is of the same level as the aggregated noise in DPPP, hence much more noise is added to local prediction compared to the noise added to local prediction in DPPP.

Standalone framework allows teachers to individually train local models on their limited local training data without any collaboration, and an end user directly sends test query to each teacher, then local prediction is released under the guarantee of \((\epsilon, \delta)\)-LDP. Therefore, this framework is expected to deliver maximum privacy, but minimum utility.

Here we remark that under the same noise mechanism, our proposed DPPP performs similarly as the centralized/global DP framework which resorts to a trusted aggregator to add noise to the aggregated predictions or vote counts, the only difference is that DPPP adopts DDP to distribute the required level of noise in global DP to multiple teachers, while ensuring privacy of each individual.

For experimentation, we investigate four different datasets. The MNIST dataset consists of a total of 70,000 handwritten digits formatted as 28x28 gray-level images, with digits locating at the center of the image.\[1\] The classification objective is to classify the input as one of 10 possible digits \([0^\circ\sim9^\circ]\). The real-world Breast Cancer dataset\[2\] 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 NSL-KDD dataset is used for intrusion detection\[3\] which contains total 125973 records with 41 features, each record is classified into two classes: anomaly and normal. We implement on both the whole NSL-KDD dataset and a smaller subset called NSL-KDD-20 (with 20% train and test data sampled from NSL-KDD dataset).

In our experiments with MNIST, we stack two convolutional layers with max-pooling and one fully connected layer with ReLUs. We use SVM model with RBF kernel for the other datasets. 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 \[13\], 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, NSL-KDD and Breast Cancer datasets respectively. To accommodate randomness during noise addition, We run each experiment for 20 times and report the average result. \( \gamma \) is set to be 1 in all figures, i.e., assuming no compromised teachers.

In particular, the results of the standalone framework are averaged over all teachers. Since \( \delta \) should be sufficiently small compared to training data size \[20\], we set \( \delta = 1/|D| \) in all the private frameworks, where \( |D| \) is the training data size. We report the prediction accuracy of student queries in Fig. 2 and Fig. 3. As evidenced by Fig. 2, DPPP outperforms both the standalone and LDP frameworks, for all datasets under varying privacy budgets \( \epsilon \in [0.01, 1] \). The centralized and distributed non-private frameworks achieve similar accuracy, indicating that the distributed non-private framework incurs negligible accuracy degradation compared with the centralized non-private framework. Moreover, DPPP yields comparable accuracy with the distributed non-private framework when \( \epsilon \geq 0.05 \) for MNIST, NSL-KDD-20 and NSL-KDD datasets, which is also comparable to PATE where each query has a low privacy budget of \( \epsilon = 0.05 \) [13].

We also notice that the impact of DPPP is more obvious on Breast Cancer dataset compared with other datasets. 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. For a large number of teachers, the aggregated predictions are accurate despite the injection of large amounts of random noise to ensure privacy.

Moreover, to investigate the impact of different \( \delta \), we vary \( \delta \in 1e^\{\{-5\sim-4\} \} \sim \{-3\sim-2\} \} \sim \{-1\} \) with fixed \( \epsilon = 1 \). As can be observed in Fig. 3, the effectiveness of our proposed DPPP persists, and for a fixed \( \epsilon \), varying \( \delta \) follows the similar trend as varying \( \epsilon \). However, we found that there is less difference with different \( \delta \), which agrees with the findings reported in \[6\]. In contrast, Fig. 2 shows that varying the value of \( \epsilon \) has larger impact on accuracy.

To understand how the number of teachers affects privacy cost, and investigate the reason behind the comparable accuracies offered by DPPP, we study the disparity of labels assigned by teachers. Since each teacher’s vote is represented by a bit vector, hence each element of the aggregated prediction vector represents the votes for each candidate label. We measure the difference in noisy votes between the most popular label and the second most popular label, i.e., the gap. If the true gap is small, the aggregated noise might change the label assigned from the first to the second. When the number of teachers \( N \) increases, the gap becomes larger.

1. http://yann.lecun.com/exdb/mnist/
2. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
3. https://www.unb.ca/cic/datasets/nsl.html
hence the topmost vote has a larger quorum, allowing for aggregation mechanisms to output the correct label in the presence of noise. That partly explains the high accuracy of MNIST, NSL-KDD-20 and NSL-KDD datasets. However, a larger $N$ implies a smaller training set allocated for each teacher, potentially reducing the teacher accuracy. For large enough $N$, each teacher will have too little training data to be accurate. Hence the accuracy of labels predicted by the ensemble is greatly dependent on the number of teachers in the ensemble: the number of teachers needs to be large enough to ensure the topmost vote with a larger quorum can counter the injection of noise, hence lowering the impact of noise on accuracy. Meanwhile, the number of teachers is also limited by a trade-off between task complexity and data availability.

Computation and Communication Overhead: We use the typical 1024-bit key size and implement a $(N, \lfloor \frac{3}{2}N \rfloor)$-threshold Paillier cryptosystem using Paillier Encryption Toolbox [21]. 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 $256^c3=768^c$ 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.

8 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 — usually considered in federated learning — 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 parameters or updates with the aggregator [14], [15], but giving the aggregator access to all teachers’ parameters or updates clearly risks privacy leakage. To privately share individual model updates, Bonawitz et al. [22] propose 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.

9 CONCLUSION AND FUTURE WORK

We present 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 and homomorphic encryption are combined to preserve individual privacy, maintain utility and ensure aggregator obliviousness. For the Binomial mechanism, we offer tighter bounds than the previous work.
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, and conduct detailed comparisons between Binomial mechanism and other DP mechanisms. We also expect to extend our framework to various machine learning scenarios beyond classification.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Benjamin Rubinstein, Dr. Justin Bedo, Dr. Chris Culnane and Prof. Vanessa Teague for the early discussions on DDP and homomorphic encryption. The authors also would like to thank Dr. Xingjun Ma for discussions about unbalanced and non-IID data distributions.

REFERENCES

[1] L. Ohno-Machado, P. S. P. Silveira, and S. 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] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction APIs.” in *USENIX Security Symposium*, 2016, pp. 601–618.

[3] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, “Membership inference attacks against machine learning models,” in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 3–18.

[4] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519.

[5] A. Blum, C. Dwork, F. McSherry, and K. Nissim, “Practical privacy: the SuLQ framework,” in *Proceedings of the twenty-fourth ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2005, pp. 128–138.

[6] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, “Deep learning with differential privacy,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[7] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, “Privacy-preserving aggregation of time-series data,” in *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[8] I. Damgård and M. 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.

[9] V. Rastogi and S. Nath, “Differentially private aggregation of distributed time-series with transformation and encryption,” in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 735–746.

[10] G. Acs and C. Castelluccia, “I have a dream(differentially private smart metering).” in *Information hiding*. vol. 6958. Springer, 2011, pp. 118–132.

[11] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, “cpsgd: Communication-efficient and differentially-private distributed sgd,” in *Advances in Neural Information Processing Systems*, 2018, pp. 7564–7575.

[12] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. 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.

[13] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar, “Semi-supervised knowledge transfer for deep learning from private training data,” *arXiv preprint arXiv:1610.05735*, 2016.

[14] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, “Federated learning of deep networks using model averaging,” *arXiv preprint arXiv:1602.05629*, 2016.

[15] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, “Learning differentially private recurrent language models,” in *ICLR*, 2018.