Federated Calibration and Evaluation of Binary Classifiers

Graham Cormode
Meta
gcormode@meta.com

Igor L. Markov
Meta
imarkov@meta.com

ABSTRACT

We address two major obstacles to practical deployment of AI-based models on distributed private data. Whether a model was trained by a federation of cooperating clients or trained centrally, (1) the output scores must be calibrated, and (2) performance metrics must be evaluated — all without assembling labels in one place. In particular, we show how to perform calibration and compute the standard metrics of precision, recall, accuracy and ROC-AUC in the federated setting under three privacy models (*i*) secure aggregation, (*ii*) distributed differential privacy, (*iii*) local differential privacy. Our theorems and experiments clarify tradeoffs between privacy, accuracy, and data efficiency. They also help decide if a given application has sufficient data to support federated calibration and evaluation.

PVLDB Reference Format:
Graham Cormode and Igor L. Markov. Federated Calibration and Evaluation of Binary Classifiers. PVLDB, 16(11): 3253 - 3265, 2023.
doi:10.14778/3611479.3611523

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://figshare.com/s/607998e479b0778645f6.

1 INTRODUCTION

Modern data management places increased focus on deploying and managing models for predicting and classifying data. Such deployed systems draw insights from large amounts of data, e.g., by training prediction models on collected labels. Traditional data workflows assemble all data in one place, but much human-generated data — the content viewed online and reactions to this content, geographic locations, the text typed, sound and images recorded by portable devices, interactions with friends, interactions with online ads, online purchases, etc — is subject to privacy constraints and cannot be shared easily [4]. This raises a major challenge: extending data processing systems to accommodate a federation of cooperating distributed clients with individually private data. In a general framework to address this challenge, instead of collecting the data for centralized processing, clients evaluate a candidate model on their labeled data, and send updates to a server for aggregation. For example, *federated training* performs learning locally on the data in parallel in a privacy-respecting way: the updates are gradients that are summed by the server and then used to revise the model [19].
Federated learning (FL) [16] can offer privacy guarantees to clients. At a baseline level of disclosure limitation, the clients never share raw data but only send out model updates. Formal privacy guarantees are obtained via careful aggregation and by adding noise to updates [30]. Given the prominence of federated learning, different aspects of the training process (usually for neural networks) have received much attention: aiming to optimize training speed, reduce communication and tighten privacy guarantees. However, deploying trained models usually requires to (*i*) evaluate and track their performance on distributed (private) data, (*ii*) select the best model from available alternatives, and (*iii*) calibrate a given model to frequent snapshots of evolving data (common in industry applications). *Strong privacy guarantees for client data during the model training must be matched by similar protections for the entire data pipeline.* Otherwise, divulging private information of clients during product use would negate earlier protections. E.g., knowing if some label agrees with model prediction can effectively reveal the private label. Concretely, Matthews and Harel [18] showed that disclosure of an ROC curve allows some recovery of the sensitive input data. In this paper, we develop algorithms for (*i*) federated evaluation of classifier-quality metrics with privacy guarantees and (*ii*) performing classifier calibration, explained further below.
Calibration and evaluation of classifiers form fundamental tasks that arise regardless of federated learning and deep learning [23]. They are important *whenever* a classifier is used in a deployed data processing system, e.g., by distributed clients on data that cannot be collected centrally due to privacy concerns. Common **evaluation** scenarios arising in practical deployments include cases such as

- A heuristic rule-based model used as a baseline for a task is evaluated to understand if a more complex ML-based solution is needed;
- A model pre-trained via transfer learning for multiple tasks (e.g., BERT [8]) is evaluated for its performance on a particular task;
- A new model trained with FL on fresh data needs to be compared with earlier models on distributed test data before launching;
- A model has been deployed to production, but its predictions must be continuously evaluated against user behavior (e.g., click-through rate) to determine when retraining is needed.

Calibration of a classifier score function remaps raw score values to probabilities, so that examples assigned probability p are positive (approximately) a p proportion of the time. Since classifier decisions are routinely made by comparing the evaluated score to a threshold, calibration ensures the validity of the threshold (especially for nonstationary data) and the transparency/explainability of the classification procedure. As noted by Guo et al. [13], “modern neural networks are not well calibrated”, prompting recent study of calibration techniques [20]. Calibration (if done well) does not affect precision-recall tradeoffs.

In this paper, we address federated evaluation of standard binary-classifier metrics: precision, recall, accuracy and ROC-AUC [23]. Accuracy measures the fraction of predictions that are correct, while recall and precision focus on only examples from a particular class, giving the fraction of these examples that are predicted correctly,

Proceedings of the VLDB Endowment, Vol. 16, No. 11 ISSN 2150-8097.
doi:10.14778/3611479.3611523

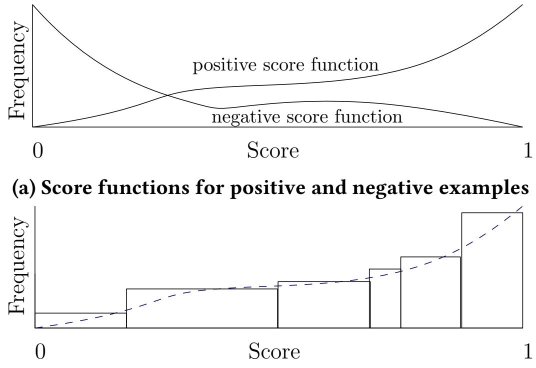
and the fraction of examples labeled as being in the class that are correct, respectively. ROC-AUC is defined in terms of the area under the curve of a plot of the tradeoff between false positive ratio and true positive ratio. These simple metrics are easy to compute when the test data is held centrally. It is more challenging to compute them in a distributed setting, while providing formal guarantees of privacy and accuracy. For privacy, we leverage secure aggregation and/or differential privacy (Section 2.2). For accuracy, we seek error bounds when estimating metrics for a given privacy requirement, so as to facilitate practical use. We present bounds as a function of the number of participating clients (M) and privacy parameters (ϵ). For the evaluation metrics we studied, errors decrease as low polynomials of M — good news for medium-to-large deployments.

The **federated computation techniques** we use are instrumental for the challenges addressed in this paper. These techniques compile statistical descriptions of the classifier score function as evaluated on the examples held by distributed clients. The description we need is a histogram of the score distribution, whose buckets divide up the examples evenly. This is attractive for several reasons. (i) Our algorithms simply estimate classifier quality, ROC AUC, and the calibration mapping by evaluating these functions on the histogram representation of the score function. (ii) Histograms are well-suited to the federated computing model, and are well-studied under different privacy models, while being robust to heteregenous data allocations. (iii) The approach is independent of the classifier type – it only needs to see the scores the classifier gives to examples.

The estimation error drops for more detailed histograms: for a B-bucket histogram, error due to the histogram scales as $1/B$ or even $1/B^2$ in some cases. Hence, $B \leq 100$ leads to very accurate results. Fortunately, the histogram-based approach is compatible with multiple privacy models that provide strong guarantees under different scenarios by adding noise to data and combining it. The chief novelty in our work is in showing *practical solutions* for these important problems via histogram computations, and in *proving accuracy bounds*. Privacy noise adds error as $M^{-1/4}$ (worst case) or M^{-1} (the best case). Hence, we obtain good accuracy (under privacy) with upwards of several thousands of participating clients.

The three models of privacy considered in this work are: (1) Federated privacy via Secure Aggregation (Federated for short), where the protocol reveals the true output of the computed aggregate (without noise addition) [24]. The secrecy of the client's inputs is achieved by using Secure Aggregation to gather their values, only revealing the aggregate (typically, the sum); (2) Distributed Differential Privacy (DistDP), where each client introduces a small amount of noise, so that the sum of all these noise fragments is equivalent in distribution to a central noise distribution such as Laplace or Gaussian, leading to a guarantee of differential privacy [11]. Using Secure Aggregation then ensures that only the sum of inputs with privacy noise is revealed; and (3) Local Differential Privacy (LocalDP), where each client adds sufficient noise to their report to ensure differential privacy on their output, so that Secure Aggregation is not needed [32]. These models imply different error bounds as we trade accuracy for the level of privacy and trust needed.

Our contributions offer algorithmic techniques and error bounds for federated calibration of classifier scores and key classifier quality metrics. These honor the three different privacy models (above).



(a) Score functions for positive and negative examples

(b) Histogram for the positive score function

Figure 1: Score functions and score histograms

For local and distributed differential privacy we use the standard ϵ privacy parameter (see Section 2.2), along with the B and M parameters (above) and the histogram parameter h explained in Section 2.3. Our theoretical bounds are summarized in Table 1, and explained in greater detail in Section 3. These rely on carefully bounding the contributions from different sources of uncertainty, then choosing parameters to minimize the sum of these uncertainties. Several key questions we address have largely eluded prior studies in the federated setting, and even modest asymptotic improvements over prior results are significant in practice due to large values of M. Compared to a prior result on ROC AUC estimation under LocalDP [3], we asymptotically improve bounds under less restrictive assumptions. Recently, heuristics have been proposed for AUC estimation based on local noise addition [26, 27]. These do not provide any accuracy guarantees, and we see that our approach provides better results in our experimental study. Similar questions have also been studied in the centralized model of DP [25], but these too lack the accuracy guarantees we can provide.

2 PRELIMINARIES

Supervised binary classification supports many practical applications, and its theoretical setting is conducive for formal analyses. It also helps address multiclass classifications and subset selection (via indicator classifiers), while lightweight ranking is routinely implemented by sorting binary classifier scores trained to predict the examples most likely to be selected. Standard classifier metrics are often approximated in practice (e.g., by Monte Carlo sampling), but this becomes more challenging in the federated setting.

2.1 Classifier Calibration and Evaluation

The input to our problem is a set of examples E. Each example x has a ground-truth label y which is either positive (+1) or negative (-1). Given a set of M examples $\mathcal{X} = \{(x_i, y_i)\}$, these are partitioned into disjoint subsets $\mathcal{P} = \{(x, +1)\}$ and $\mathcal{N} = \{(x, -1)\}$ of positive and negative examples, of size P and N respectively. Given a trained score function $w(\cdot)$ which takes in examples x and outputs a score $w(x) \in [0, 1]$, we define a binary classifier based on a threshold T, as $\text{pred}(x) = \mathbb{I}[w(x) \geq T]$. At $T = 0$, the *false positive ratio* (FPR) and *true positive ratio* (TPR) are 1, and drop to 0 as $T \longrightarrow 1$.

Let $p(s)$ and $n(s)$ denote the score (mass) functions p and n of the positive and negative examples respectively. That is, given a set of positive examples $\mathcal{P}$, $p(s) = |\{w(x) < s | x \in \mathcal{P}\}|/P$ and $n(s) = |w(X) < s | x \in \mathcal{N}\}|/N$ for negative examples $\mathcal{N}$. Figure 1a shows an example of these functions: positive examples tend to have higher scores, while the negative examples have lower scores.

Well-behaved score distributions. For arbitrary score distributions, strong bounds for our problems may not be possible. But empirically significant cases often exhibit some type of smoothness (moderate change), except for point spikes – score values repeated for a large fraction of positive or negative examples: (*i*) for classifiers with a limited range of possible output scores, (*ii*) when some inputs repeat verbatim many times, (*iii*) when a dominant feature value determines the score. We call such score distributions *well-behaved*. Formally, we define a spike as any point with probability mass $> \phi$, hence at most $1/\phi$ spikes exist.

Definition 1. Spikes are points s for which $p(s) > \phi$ or $n(s) > \phi$. The score distribution is (ϕ, ℓ)*-well-behaved* if it is ℓ- Lipschitz between spikes:

$$|p(s) - p(s+\Delta)| \leq \ell\Delta \quad \text{and} \quad |n(s) - n(s+\Delta)| \leq \ell\Delta \qquad (1)$$

This smoothness condition for a parameter ℓ captures the idea that the amount of positive and negative examples does not change too quickly with s (barring spikes in $[s, s+\Delta]$).

Balanced input assumption. Our proofs spell out the dependence on the number of positive and negative examples. To simplify presentation, we may sometimes assume that counts of positive and negative examples (P and N), are each at least a constant fraction of the total number of examples $M = P+N$. Our proofs hold regardless, but the simplified claims expose core dependencies between results. Indeed, in some cases, $P = N$ (perfect balance) maximizes our error bounds and yields worst-case behavior. We can trivially assume that classifier accuracy is > 0.5. So for balanced inputs, there is a constant fraction of true positives.

Calibration. Given a score function s, *calibration* defines a transformation of s to obtain an accurate estimate of the probability that the example is positive. That is, for a set of examples and labels (x_i, y_i), we want a function $c(\cdot)$, so that $c(w(x_i)) \sim \Pr[y_i = 1]$. There are many approaches to find such a mapping c, such as isotonic regression, or fitting a sigmoid function. A baseline approach is to perform histogram binning on the function, with buckets chosen based on quantile boundaries. More advanced approaches combine information from multiple histograms in parallel [22]. To measure the calibration quality, expected calibration error (ECE) arranges the predictions for a set of test examples into a fixed number of bins and computes the expected deviation between the true fraction of positives and predicted probabilities for each bin[1].

Precision, Recall, Accuracy. Given a classifier that makes (binary) predictions $\text{pred}(x_i)$ of the ground truth label y_i on M examples x_i, standard classifier quality metrics include:

- **Accuracy**: the fraction of correctly predicted examples, i.e., $|\{i : \text{pred}(x_i) = y_i\}|/M$.
- **Recall**: the fraction of correctly predicted positive examples (a.k.a. true positive ratio), i.e., $|\{i : \text{pred}(x_i) = y_i = 1\}|/|\{i : y_i = 1\}|$.
- **Precision**: the fraction of examples labeled positive that are labeled correctly, i.e., $|\{i : \text{pred}(x_i) = y_i = 1\}|/|\{i : \text{pred}(x_i) = 1\}|$.

ROC AUC (Receiver Operating Characteristic Area Under the Curve) is often used to capture the quality of a trained ML classifier. Plotting TPR against FPR generates the ROC curve, and the ROC AUC (AUC for short) represents the area under this curve. The AUC can be computed in several equivalent ways. Given a set of labeled examples $E = \{(x, y)\}$ with ±1 label y, the AUC equals the probability that a uniformly-selected positive example ($y = 1$) is ranked above a uniformly-selected negative example ($y = -1$). Let N be the number of negative examples, $|\{(x, -1) \in E\}|$, and $P = |\{(x, 1) \in E\}|$. Then [14]

$$\text{AUC} = \tfrac{1}{PN} \textstyle\sum_{(x,1)\in E} \sum_{(z,-1)\in E} \mathbb{I}[w(x) > w(z)] \qquad (2)$$

This expression of AUC implies that we should compare pairs of examples across clients. We avoid such distributed interactions by evaluating it via histograms for guaranteed approximations.

2.2 Privacy Models

Federated Privacy via Secure Aggregation (Federated) assumes that each client submits one (scalar or vector) value $x^{(i)}$. Our protocols easily handle the case when clients hold multiple values, by working with a vector representing the aggregated inputs. The Secure Aggregation protocol computes the sum, $X = \sum_{i=1}^{M} x^{(i)}$ without revealing any intermediate values. Various cryptographic protocols provide distributed implementations of Secure Aggregation [3, 24], or aggregation can be performed by a trusted aggregator, e.g., a server with secure hardware [33]. While secure aggregation provides a practical and simple-to-understand privacy primitive, it does not fully protect against a knowledgeable adversary. In particular, knowing the inputs of all clients except for one, the adversary can subtract the values that they already hold. Hence, differential privacy guarantees are sought for additional protection.

Distributed Differential Privacy (DistDP). The model of differential privacy (DP) represents a data processing task as a randomized algorithm, so that so for two inputs that are close, their outputs have similar probabilities (within an $\exp(\epsilon)$ factor)–in our case, captured as inputs that vary by the addition or removal of one example (event-level privacy) [11]. DP is often achieved by adding calibrated noise from a suitable statistical distribution to the exact answer. In our setting, this is instatiated by introducing distributed noise at each client which sums to give discrete Laplace noise with variance $O(1/\epsilon^2)$ [2]. The resulting (expected) absolute error for count queries is $O(1/\epsilon)$, and for means of M values is $O(1/\epsilon M)$ [11, Theorem 3.8]. The distributed noise is sampled from the difference of two (discrete) Pólya distributions [2], which neatly combines with secure aggregation so only the noisy sum is revealed[2].

Local Differential Privacy (LocalDP). LocalDP can be understood as applying the DP definition at the level of an individual client: each client creates a public report based on the input that they hold (e.g, an example), so that whatever inputs the client holds,

[1] Formally, the ECE is defined by Naeini et al. [22] as $\sum_{j=1}^{K} P(j)\,|o(j) - e(j)|$, where $P(j)$ is the (empirical) probability that an example falls in the jth bucket (out of K), while $o(j)$ is the true fraction of positive examples that fall in the jth bin, and $e(j)$ is the fraction predicted by the model.

[2] Other privacy noise is possible via Binomial [9] or Skellam [1] noise addition.

the probability of producing each possible report is within an $\exp(\epsilon)$ factor [32]. When a set of M clients each hold a binary value b_i, we can estimate the sum and the mean of b_i under ϵ-LDP by applying randomized response [29]. For the sum over M clients, the variance is $O(M/\epsilon^2)$, and so the absolute error is proportional to $\sqrt{M}/\epsilon$. After rescaling by M, the variance of the mean is $O(1/(M\epsilon^2))$, and so the absolute error is proportional to $1/\epsilon\sqrt{M}$ [32]. The same bounds hold for histograms via "frequency oracles", when each client holds one out of B possibilities — we use Optimal Unary Encoding (OUE) [28] to build a (private) view of the frequency histogram.

2.3 Building Score Histograms

A key part of our algorithms is to form an equi-depth histogram of the clients' information. That is, given M samples as scalar values, we seek a set of boundaries that partition the range into B buckets with (approximately) equal number of samples per bucket. This is a well-studied problem, so in what follows we review how such histograms can be computed under each privacy model and analyze their accuracy as a function of parameters h and ϵ. The novelty in our work is using them to privately measure classifier quality with accuracy guarantees across a range of federated models.

To represent the distribution of scores with histograms, we create a set of B buckets that partition $[0, 1]$. We will build two histograms, n and p, where p_i and n_i give the number of positive and negative examples respectively whose score falls in bucket i. Figure 1b shows a histogram for the positive score function from Figure 1a. In what follows, we set the histogram bucket boundaries based on the (approximate) quantiles of the score function for the given set of examples E, so that $p_i + n_i \leq (P + N)/B$, where P and N denote the total number of positive and negative examples respectively. We formalize the problem of equi-depth histogram as follows. Given M examples with real values $z_i \in [0, 1]$ we seek $r_0 \dots r_B$, so that

$$\forall 1 \leq j \leq B : r_{j-1} < r_j, \text{ and } |\{i : r_{j-1} < z_i \leq r_j\}| = M/B. \quad (3)$$

This definition can be naturally generalized for multiple examples with the same value, and to tolerate some approximation factor.

In the central non-private setting, it is straightforward to find the exact bucket boundaries for a score histogram with equal-weighted buckets: gather all the input data points and sort them, then read off the value after every M/B examples. This is more challenging for distributed private data, but has been studied when finding the quantiles of the input data [5, 12, 31]. In what follows, we outline finding quantiles under different models of privacy and give the accuracy bounds that result. Most federated techniques gather information on the data at a suitably fine granularity, and use this information to find appropriate bucket boundaries. For a parameter h ("hierarchy height"), and for each integer k ($1 \leq k \leq h$) we divide the range $[0, 1]$ into 2^k uniform segments, each of length 2^{-k}. For each segment, we count how many data points reside in that segment. This immediately lets us answer a *prefix query* up to a granularity of 2^{-h}: given a range $[0, r/2^h]$ for an integer $r < 2^h$, we can greedily use the computed counts to partition the prefix into at most h segments, at most one for each length 2^{-k}, for $1 \leq k \leq h$. To find the point q s.t. $|\{i : z_i < q\}| = \phi M$, we perform binary search on r in order to the prefix whose sum is closest to ϕM. We now apply this idea for each of the privacy models.

Secure Aggregation. The secure aggregation case is most straightforward, since we do not introduce any privacy noise. Instead, each client can encode their data into h one-hot vectors of length 2^k for $1 \leq k \leq h$. This allows the aggregator to find a set of bucket boundaries $\hat{r}_i$ based on (3), up to $|\hat{r}_i - r_i| \leq 2^{-h}$. When the score distribution is (ϕ, ℓ)-well-behaved, the mass of client data points varies by at most ℓ per unit, meaning that the error in the number of points in this approximation is at most $\ell 2^{-h}$. We can then ensure that the parameter h is chosen so that the error in a bucket, $\ell 2^{-h}$, is at most a small fraction of the desired amount (say, $1/4$) of the bucket weight, which is $1/B$. Rearranging, we require $h > \log_2(4B\ell) = 2 + \log_2 B + \log_2 \ell$. In other words (treating ℓ also as a constant), we only require that h be $\log_2 B$ plus a constant.

Distributed DP. In the DistDP case, the aggregator obtains the data from the clients, each of whom add a small amount of noise that collectively sums up to be equal in distribution to some global noise value. In our setting, we will make use of Pólya noise which sums to the (discrete) Laplace distribution, a.k.a., the symmetric geometric distribution [2]. We can adopt the same depth h hierarchy as before, but now we have discrete Laplace noise added to every count (Section 2.2). To ensure differential privacy, the noise has to be scaled as a function of ϵ/h to achieve an overall ϵ-DP guarantee, since each client contributes to h counts. Equivalently, we could divide the clients into h batches, so that each batch reports on a single level of the hierarchy, and adds noise as a function of ϵ. In either case, the total variance of finding the number of clients in a range is $O(h^3/\epsilon^2)$, since $O(h)$ counts each with variance $O(h^2/\epsilon^2)$ are combined. This ensures that the end points for bucket boundaries can be found with expected absolute error $O(h^{3/2}/\epsilon M + 2^{-h})$. As before, the 2^{-h} term comes from representing input points at this granularity. If we balance these terms, for constant ϵ, and M between 10^3 and 10^6, we would expect to choose values of $10 \leq h \leq 20$.

Local DP. The case of local DP is somewhat similar to the distributed DP case. Here, the privacy noise is added by each client independently (typically by a version of randomized response) so that their input is encoded in a frequency oracle [28]. Standard approaches apply asymmetric randomized response [29] to a one-hot encoding of the client's input value, such as the Optimal Unary Encoding approach we use in our experiments [28]. If a client has no data to report (e.g., when building a histogram on positive examples, and the client's example is negative), the client can submit an all-zeros vector to the LDP mechanism, and obtain the same LDP guarantee for their contribution. Each client introduces noise of $O(1/\epsilon^2)$ on each count they report, and clients are divided into h groups of size M/h to report on one level of the hierarchy. Now the variance for the fraction of clients in a range is $O(h^2/(\epsilon^2 M))$, due to the increased noise level. This means that bucket end points are found with expected absolute error $O(h/\epsilon\sqrt{M} + 2^{-h})$, so a shallower hierarchy is preferred: for typical parameter settings, we now expect $5 \leq h \leq 10$.

Overcoming heterogeneity. A common concern when working in the federated model is *data heterogeneity*: values held by clients may be non-iid (some clients are more likely to have examples of a single class), and some clients may hold more examples than others. By working with histogram representations we *overcome* these concerns: the histograms we build are insensitive to how the data is

Table 1: Our simplified error bounds in three privacy models

	Federated	DistDP	LocalDP
P/R/A (for a score function)	$1/B$	$1/\epsilon^{2/3}M^{2/3}$	$1/\epsilon^{2/3}M^{1/3}$
ROC AUC	$1/B^2$	$(\frac{1}{\epsilon}+\frac{1}{B})\frac{1}{M}$	$h/\epsilon M^{1/2}$
Expected Calibration Error	$1/M^{1/3}$	$1/M^{1/3}$	$1/\epsilon^{1/2}M^{1/4}$

distributed to clients, and the privacy noise needed is *independent of* heterogeneity. We can then state results in terms of a few basic parameters (number of clients M and histogram buckets B etc.), and independent of other properties of the data distribution.

3 SUMMARY OF OUR RESULTS

Table 1 presents simplified versions of our main results as a function of the parameters introduced above. Here and throughout we express asymptotic error bounds in terms of the *expected absolute error*, which is also a bound that holds with high probability via standard concentration inequalities [21]. Without requiring any i.i.d. assumptions for distributed clients, our results clarify the expected magnitude of the error, which should be small in comparison to the quantity being estimated: tightly bounding the error values ensures accurate results. All the estimated quantities are in the $[0, 1]$ range. For (binary) classifiers of interest the four quality metrics will be $\geq \frac{1}{2}$, while the expected calibration error is a small value in $[0, 1]$.

To keep the presentation of these bounds simple, we use the *balanced input assumption* (from Section 2.1), i.e., there are $\Theta(M)$ positive examples and $\Theta(M)$ negative examples among the M clients[3]. Error bounds are presented as a function of the number of clients, M, the privacy parameter ϵ, the number of buckets used to build a score histogram, B, and the hierarchy height, h (see above). Across the various problems the error increases as we move from Federated to DistDP to LocalDP. This is expected, as the noise added in each case increases to compensate for the weaker trust model.

Other trends we see are not as easy to guess. Increasing the number of buckets B often helps reduce the error, but this is not always so, particularly for the LocalDP results. Increasing the number of examples, M, typically decreases the error, although the rate of improvement as a function of M varies from $1/M^{1/4}$ in the worst case to $1/M$ in the best case. Our experimental findings presented in Section 7 agree with this analysis and confirm the anticipated impact of increasing M and of varying the parameters B and h. We observe high accuracy in the Federated case and good accuracy when DP noise is added. Calibration error for DistDP is insensitive to ϵ as explained after Theorem 8. These results help building full-stack support for practical federated data processing, and show the practicality of federated classifier evaluation.

4 PRECISION, RECALL, AND ACCURACY

Overview. Given a score function $w(\cdot)$ with values in $[0, 1]$ for each example x, we seek statistics that would help estimate the *precision*, *recall* and *accuracy* of the classifier defined by a threshold T, where T can be chosen at query time. Our solution is to build score histograms of the positive and negative examples (via Section 2.3), and compute the metrics using just the histograms. That is, we break each calculation into a sum over histogram buckets. We can bound the uncertainty due to this approximation by limiting the number of examples in the bucket, and bound the uncertainty due to privacy noise by limiting the number of buckets. We balance these two sources of uncertainty to determine the optimal number of buckets as a function of privacy ϵ and number of client values M. We will use the following mathematical fact.

Fact 1. *For (positive) real numbers A and G, suppose we are given $\hat{A} = A \pm \alpha$ and $\hat{G} = G \pm \gamma$, where $\gamma \leq G/2$ and $\hat{X} = X \pm x$ is shorthand for $\hat{X} \in [X - x, X + x]$. When estimating a fraction $\frac{A}{G} \leq 1$, we have $|\frac{\hat{A}}{\hat{G}} - \frac{A}{G}| = O(\frac{\alpha+\gamma}{G})$.*

Proof.

$$\left|\frac{A\pm\alpha}{G\pm\gamma} - \frac{A}{G}\right| = \left|\frac{A\pm\alpha}{G(1\pm\gamma/G)} - \frac{A}{G}\right| = \left|\frac{(A\pm\alpha)(1\mp2\gamma/G)}{G} - \frac{A}{G}\right|$$
$$\leq \frac{2A\gamma}{G^2} + \frac{\alpha}{G} + \frac{2\alpha\gamma}{G^2} = O\left(\frac{\alpha+\gamma}{G}\right) \qquad (4)$$

using $A/G \leq 1$ and $\gamma/G \leq 1/2$ to simplify in the final step. □

Theorem 2. *Given a score histogram for positive and negative examples built based on a hierarchy of height h, we can compute estimates for precision, recall and accuracy based on a threshold T which approximate the true precision, recall and accuracy for a threshold $T' \in T \pm \Delta$, under the balanced input assumption, as follows:*

- *In the basic Federated case, we achieve error $O(1/B)$ with $\Delta = 2^{-h}$;*
- *For DistDP, the error is $O(1/(\epsilon M)^{2/3})$ with $\Delta = O(h^{3/2}/\epsilon M + 2^{-h})$;*
- *For LocalDP, error is $O(1/(\epsilon^{2/3}M^{1/3}))$ with $\Delta = O(h/\epsilon\sqrt{M} + 2^{-h})$.*

Proof. We make use of score histograms of positive and negative examples that are accurate up to a small uncertainty in T, which we write as $(T \pm \Delta)$. The histogram takes a parameter h that determines the height of the hierarchy used to construct it. Under the Federated model, we have that $\Delta = 2^{-h}$, whereas for DistDP $\Delta = O(h^{3/2}/\epsilon M + 2^{-h})$ and for LocalDP $\Delta = O(h/\epsilon\sqrt{M} + 2^{-h})$, as explained in Section 2.3. We consider each classifier metric in turn. **Accuracy** is easiest to handle, since we just need the numerator

$$|\{i : y_i = -1 \wedge w(x_i) < T\}| + |\{i : y_i = 1 \wedge w(x_i) \geq T\}|$$

That is, the number of negative examples with a score below T plus the number of positive examples with a score of at least T. We can estimate both these quantities with additive error at most $1/B$ using a B-bucket equi-depth histogram (without noise addition).

In the DistDP case, there is discrete Laplace noise on each bucket count to mask the presence of any individual. We can bound the error from this noise to be of order $\sqrt{B}/\epsilon M$, by summing variances, giving a total error bound of $O(1/B+\sqrt{B}/\epsilon M)$. We can balance these two terms so that $1/B = \sqrt{B}/\epsilon M$, so $B^3 = \epsilon^2 M^2$, i.e., $B = (\epsilon M)^{2/3}$. This gives the total error as $O(1/(\epsilon M)^{2/3})$.

Under ϵ-LocalDP noise, we obtain an additional error term of $\sqrt{B}/\epsilon\sqrt{M}$ (summing the variance over B buckets). Balancing these two terms sets $1/B = \sqrt{B}/\epsilon\sqrt{M}$, i.e., $B^3 = \epsilon^2 M$, and so $B = \epsilon^{2/3}M^{1/3}$. Under this setting, the total error is bounded as $O(1/\epsilon^{2/3}M^{1/3})$.

Recall. The same histogram approach works for recall. Using a histogram, we aim to estimate the number of true positives, which is the number of positive examples above the threshold, divided by the total number of positives. Without DP noise, we can compute P, the number of positive examples, exactly for the denominator, but

[3] The results without this assumption appear in the proofs of the respective claims.

we incur error M/B for the numerator, giving error M/BP. To simplify this expression, we can invoke the *balanced input* assumption, which bounds this by $O(1/B)$ since $M = O(P)$.

Including LocalDP noise, we incur error $\alpha = \sqrt{BM}/\epsilon$ when summing over B buckets. We also have error $\gamma = \sqrt{M}\epsilon$ for estimating P. Using Fact 1, the error is dominated by $O(\sqrt{BM}/\epsilon P + M/BP)$. This is again balanced by setting $B = \epsilon^{2/3}M^{1/3}$. Likewise, for DistDP noise, $\alpha = \sqrt{B}M/\epsilon$ and $\gamma = M\epsilon$, which fixes $B = (\epsilon M)^{2/3}$ for error $O(1/B) = O((\epsilon M)^{-2/3})$ under the balanced input assumption.

Precision. The bounds for precision are similar. Using a histogram, we want to first count how many examples are correctly classified as positive – this is the number of positive examples above the threshold T. We scale this by the total number of examples that are classified as positive, which is just the number of examples above the threshold T. Under secure aggregation, we can estimate both of these with error M/B, which is due to the histogram bucketing. Plugging these into (4), the error bound is $O(M/B(TP + FP))$. To simplify the form of this bound, we invoke the balanced input assumption. This implies that a constant fraction of the examples are positives, and that the classifier has at least a constant accuracy, so the bound becomes $O(1/B)$.

With LocalDP noise, both of these quantities incur additional noise of $\alpha = \gamma = \frac{\sqrt{BM}}{\epsilon}$. The error bound is $O(\frac{1}{(TP+FP)}(\frac{M}{B} + \frac{\sqrt{BM}}{\epsilon\sqrt{M}}))$. Balancing this error sets $B = \epsilon^{2/3}M^{1/3}$, which gives an error of $O(M^{2/3}/\epsilon^{2/3}(TP+FP))$. If $FP+TP$ is a constant fraction of M, then we simplify this to $O(1/\epsilon^{2/3}M^{1/3})$. Similarly for DistDP noise, we have $\alpha = \gamma = \sqrt{B}M/\epsilon$, which leads to error $O(1/(\epsilon M)^{2/3})$ under the same assumptions on positive examples.

Combining each of these bounds with the error introduced by using a score histogram to find the bucket boundaries on the threshold as Δ, we obtain the results stated in the theorem. □

In the Federated case, accuracy improves without limit if we increase the height of the hierarchy h arbitrarily and scale $B \sim 2^h$. The cost is that the resulting histogram built by the aggregator is $O(2^h)$ in size. However, for DistDP and LocalDP, increasing h increases the imprecision Δ: there is uncertainty due to the privacy noise, which eventually outweighs the fidelity improvement due to smaller histogram buckets. Our analysis for Theorem 2 balances the two terms to find a setting of B that yields the stated bounds.

5 FEDERATED COMPUTATION OF ROC AUC

Overview. Estimating ROC AUC is a fundamental problem in classifier evaluation. It is particularly challenging in the federated setting, since it requires comparing how different examples are handled by the classifier, whereas these examples are usually held by different clients. However, it turns out that we can get accurate approximations of AUC without requiring communication amongst clients. Our approach is to represent the positive and negative score distributions with B-bucket score histograms, and use these to compute the AUC. That is, we treat the piecewise-constant function (Figure 1b) as if it were the true score function. We first show that this approach has bounded error in the federated case (Section 5.1), then tighten this under our well-behaved assumption (Section 5.2. We then give the corresponding results for the DistDP and LocalDP models (Section 5.3 and 5.4).

In more detail, we use B-bucket score histograms (Section 2.3) to write a histogram-based estimator for AUC:

$$H_B = \frac{1}{PN} \sum_{i\in[B]} \left(\sum_{j<i} p_i n_j + \frac{1}{2} p_i n_i\right) \qquad (5)$$

Recall that P and N denote the total number of positive and negative examples, while p_i and n_i denote the number of positive and negative examples in bucket i of the histogram.

We start by analysing the accuracy when the histogram contains exact counts, i.e., in the Federated case. Compared to the precise AUC computation, our uncertainty in this estimate derives from the $p_i n_i$ term: for any $j < i$, we know that all the pairs of examples that contribute to $p_i n_j$ would be counted by (2), while for $j > i$, no pairs corresponding to $p_i n_j$ should be counted. However, within bucket i, we are uncertain whether all positive examples are ranked higher than all negative examples (in which case we should count $p_i n_i$ towards (2)), or vice-versa (yielding a zero contribution). The choice of $\frac{1}{2}p_i n_i$ in (5) takes the midpoint of these two extremes. Section 5.2 shows this is a good choice for well-behaved distributions.

5.1 Worst-case Bounds via Score Histograms

We first present a general bound on AUC estimation using a score histogram. A key insight is that, in the Federated case, the only uncertainty comes from the contribution to the AUC of positive and negative examples that fall in the same bucket. Using an equi-depth histogram bounds the number of such examples, and so the absolute error drops as the number of histogram buckets grows.

Lemma 3. *In the Federated case, the additive error in AUC estimation with a B-bucket score histogram is $O(1/B)$.*

Proof. In order to bound the error in our estimate of AUC, we can choose the histogram buckets based on the quantiles of the score function (Section 2.3), so that $p_i + n_i = (P + N)/B$. The error in our estimate is at most $\frac{1}{2PN} \sum_{i\in[B]} p_i n_i$. This error is maximized when $p_i = n_i = (P + N)/2B$, so that the resulting (absolute) error is

$$\frac{1}{2PN} \sum_{i\in[B]} p_i n_i = \frac{1}{2PN} \sum_{i\in[B]} \frac{(P+N)^2}{(2B)^2} = \frac{4}{2(P+N)^2} \frac{B(P+N)^2}{4B^2} = \frac{1}{2B}$$

using that our *worst-case* setting of p_i and n_i entailed that $P = N = (P + N)/2$ (the perfectly balanced input case). □

The proof considers worst-case allocations of examples with a uniform share of positive and negative examples in each bucket, to show that using a histogram with B buckets set by the quantiles of the score function suffices to bound the (additive) AUC error by $1/2B$. I.e., $B = 50$ buckets ensure that the error ≤ 0.01. For a classifier with AUC $> \frac{1}{2}$, this means a relative error of at most $1/B$.

5.2 Better Bound for Well-behaved Distributions

The worst-case bound allows extreme cases where all positive examples in a bucket are ranked above all negative examples in the same bucket, or vice-versa. We show a tighter bound when the distribution functions of examples are *well-behaved* (Defn. 1).

Theorem 4. *For $(1/B, \Theta(1))$-well-behaved inputs, the Federated error for AUC estimation using score histograms is $O(1/B^2)$.*

Proof. We consider the maximum uncertainty we can have within a single histogram bucket i when the score distribution is

$(1/B, \Theta(1))$-well-behaved. First, suppose that there is a spike within the bucket. Choosing our spike parameter $\phi = 1/B$, we have that the bucket must contain *only* this spike, otherwise the bucketing would violate the quantile property (3). Thus, we have no uncertainty as to the contribution of the single point x in this bucket to the AUC, as it is zero according to (2). Hence, providing the (ϕ, ℓ)-well-behaved property holds for $\phi = 1/B$, we incur no error due to spikes.

This leaves only buckets without spikes, which are assumed to obey the Lipschitz condition with parameter $\ell = \Theta(1)$. Abusing notation slightly, let p_i and n_i denote the mass of positive and negative examples at the left hand end of the bucket. We reparameterize the mass function within a bucket based on a parameter $0 \leq \alpha \leq 1$, so that $p(\alpha)$ and $n(\alpha)$ give the mass of examples within the bucket at the point that is an α fraction across the bucket (from left to right). We define $L = \ell\Delta_i$, where Δ_i is the width of bucket i. Then

$$|p(\alpha) - p_i| \leq L\alpha \qquad \text{and} \qquad |n(\alpha) - n_i| \leq L\alpha$$

Rearranging, we can write $p(\alpha) \in p_i \pm L\alpha$ and $n(\alpha) \in n_i \pm L\alpha$.

The contribution to AUC from this bucket is then bounded by integration of these linear bounding functions:

$$\begin{aligned}\int_0^1 p(\alpha) \int_0^\alpha n(\alpha')d\alpha' d\alpha &\in \int_0^1 (p_i \pm L\alpha) \int_0^\alpha (n_i \pm L\alpha')d\alpha' d\alpha \\ &\in \int_0^1 (p_i \pm L\alpha) \left(n_i\alpha \pm \tfrac{L\alpha^2}{2}\right) d\alpha \\ &\in \left[\tfrac{p_i n_i \alpha^2}{2} \pm \tfrac{L n_i \alpha^3}{3} \pm \tfrac{p_i L \alpha^3}{6} \pm \tfrac{L^2\alpha^4}{8}\right]_0^1 \\ &\in \tfrac{1}{2}p_i n_i \pm \tfrac{1}{3}L n_i \pm \tfrac{1}{6}L p_i \pm \tfrac{1}{8}L^2.\end{aligned}$$

If we use $p_i n_i/2$ as our estimate of the contribution to the AUC from bucket i, the absolute error in this estimate is at most $Ln_i/3 + Lp_i/6 + L^2/8 = O(\Delta_i(p_i + n_i) + \Delta_i^2)$ (treating the Lipschitz parameter ℓ as a constant). Without loss of generality, we can assume that $\Delta_i \leq 1/B$ – the width of any bucket is at most $1/B$. Although this is not directly implied if we define the buckets by the quantiles of the score functions, we can additionally enforce this property without changing that there are $O(B)$ buckets in the histogram. The uncertainty in AUC contribution is $O((p_i + n_i)/B + 1/B^2) = O((P + N)/B^2)$, from our choice of bucket boundaries and the bounded number of points in a bucket. Summing over all buckets and normalizing by $1/PN$ the absolute error in AUC is bounded by $O(\frac{P+N}{BPN})$ which simplifies to $O(1/BM)$ under the balanced input assumption. To express this solely in terms of B, we can observe that $B < N$ and $B < P$ (else, we have empty buckets, which don't contribute to the error), and so the error bound is $O(B^{-2})$. □

This improved $1/B^2$ scaling is strong and produces tight error bounds for small B. Picking $B \approx 100$ gives error $\approx 10^{-4}$, small enough for most conceivable applications. Empirical errors on AUC estimates closely follow $O(B^{-2})$ on test data (Section 7).

5.3 AUC Noise Addition for DistDP

Differential privacy introduces noise into every bucket, which quickly becomes the dominant factor.

Theorem 5. *Under DistDP, the AUC estimation error bound with the* balanced input assumption *is* $O((\frac{1}{\epsilon} + \frac{1}{B})\frac{1}{M})$.

Proof. Under differential privacy, we additionally have to account for privacy noise on the counts. We first consider the effect of centralized DP noise added to each histogram bucket. Recall that, as described in Section 2.2 the effect of the ϵ-DP noise is to add unbiased noise of variance $O(1/\epsilon^2)$ (i.e., with magnitude $\Theta(1/\epsilon)$) to every count. This means that there are errors introduced in the estimates of $p_i n_j$, as well as $p_i n_i$.

Errors also arise due to the variation in the size of histogram buckets: if we estimate quantiles under differential privacy, then we no longer guarantee that there are exactly M/B examples in each histogram bucket. However, the analysis is not highly sensitive to this issue, and it suffices to assume that the private histogram guarantees that there are between $M/2B$ and $2M/B$ examples in each bucket. This is the case using the DistDP histogram construction of Section 2.3 for typical choices of the parameters h, ϵ, M and B (say, M more than a few hundred). We make use of the expression for the variance of the product of two independent random variables,

$$\mathsf{Var}[XY] = \mathsf{Var}[X]\mathsf{Var}[Y] + \mathsf{Var}[X](\mathsf{E}[Y])^2 + \mathsf{Var}[Y](\mathsf{E}[X])^2. \quad (6)$$

We apply this expression to the estimate of $p_i n_j$, since the random variables representing privacy noise on each of p_i and n_j are truly independent. The total variance in the use of a noisy histogram with B buckets, $\hat{H}_B$, in (5) to approximate (2) is given by

$$\begin{aligned}O\left(\textstyle\sum_{i,j} \frac{1}{\epsilon^4} + \frac{p_i^2}{\epsilon^2} + \frac{n_j^2}{\epsilon^2}\right) &= O\left(\frac{B^2}{\epsilon^4} + B\textstyle\sum_{i\in[B]} \frac{p_i^2 + n_i^2}{\epsilon^2}\right) \\ &= O\left(\frac{B^2}{\epsilon^4} + \frac{P^2 + N^2}{\epsilon^2}\right)\end{aligned}$$

This expression is dominated by the quadratic terms in P and N for ϵ at least a constant, i.e., we can use $O((P + N)^2 (1/\epsilon^2))$ as a bound on the variance, since we can assume $P > B$ and $N > B$. Combining the error bound from Theorem 4 and after normalizing by the factor of PN, this yields an absolute error of magnitude $O((\frac{1}{\epsilon} + \frac{1}{B})\frac{P+N}{PN})$, i.e., augmenting $1/B$ from the noiseless case with an additional $1/\epsilon$. Under the *balanced input assumption*, we can write the total error bound as $O((\frac{1}{\epsilon} + \frac{1}{B})\frac{1}{M})$. □

That is, the error is comprised of two components: privacy noise of $O(1/\epsilon M)$, and "bucketization" noise of $O(1/BM)$. Since ϵ can usually be treated as fixed, this rules out asymptotic benefit for increasing B above $\Theta(\epsilon)$: when B is large enough, the error due to privacy noise will dominate, and using more buckets will not help. Empirical results in Section 7.2 confirms this: for $\epsilon = 1$, $B \gg 20$ makes negligible difference in terms of accuracy.

5.4 AUC Noise Addition for LocalDP

The LocalDP case is similar, except the magnitude of the noise is larger, since we incur noise on every example. Here, the error of the quantile estimates is also larger, but this does not affect things.

Theorem 6. *The error bound for LocalDP AUC estimation with the* balanced input assumption *with a hierarchy height h is* $O(h/\epsilon\sqrt{M})$.

Proof. As in the DistDP case, we assume that M is large enough so the error from determining the quantile boundaries is sufficiently small that each bucket has a constant multiple of M/B examples in it. This means that $h/\epsilon\sqrt{M} \leq M/B$. Rearranging, we require $Bh/\epsilon = O(M^{3/2})$. For constant ϵ, and typical bounds $h \leq 20$, $B \leq 10^3$, this will hold provided that M is in the order of thousands or more.

Let V_ϵ denote the variance when using the LDP frequency oracle to answer a range query (the sum of a range of buckets in the histogram). Prior work has determined that when we use a hierarchical histogram [5] of height h with Optimal Unary Encoding, $V_\epsilon = \frac{h^2 \exp(\epsilon)}{(\exp(\epsilon)-1)^2}$. Let $M = N + P$ denote the total number of examples. Considering the variance in the estimation of $\sum_{i\in[B]} \sum_{j<i} p_i n_j$ via (6), we can write

$$\begin{aligned}\mathsf{Var}[\textstyle\sum_{i\in[B]} \sum_{j<i} p_i n_j] &= O\left(\textstyle\sum_{i,j} M^2 V_\epsilon^2 + p_i^2 M V_\epsilon + n_j^2 M V_\epsilon\right)\\ &= O\left(\textstyle\sum_{i,j} M^2 V_\epsilon^2 + M^3 V_\epsilon / B^2\right)\\ &= O\left(M^2 B^2 V_\epsilon^2 + M^3 V_\epsilon\right)\end{aligned}$$

For small B, the term in M^3 will dominate. If we balance the two terms, we obtain $B = O(\sqrt{M/V_\epsilon})$.

For $\epsilon = O(1)$, we have that $V_\epsilon = O(h^2/\epsilon^2)$. Consequently, the absolute error is of magnitude $O(\sqrt{V_\epsilon/M}) = O(h/\epsilon\sqrt{M})$. That is, the dependence on $M = P + N = O(\sqrt{PN})$ is weakened, so error decreases more slowly as the number of examples increases. □

We compare this bound to a result of Bell et al. [3] where a bound of $O(h^{3/2}/\epsilon\sqrt{M})$ is derived for LDP AUC estimation. Their setting assumes a discrete domain with 2^h possible values and non-private classes of the examples, whereas we remove those assumptions.

6 FEDERATED SCORE CALIBRATION

Overview. For calibration, we will look at the histogram of the score functions, and apply the calibration to this representation. That is, we will compute a calibrated score for each bucket, based on the observed number of positive and negative examples in that bucket. Classifier calibration poses an additional challenge, since the quality of a calibrated classifier is determined by its performance averaged over multiple examples. When building a summary from a histogram representation of the labeled data, we incur additional uncertainty: for small buckets holding a few points, our estimates of classifier metrics within such buckets are noisier and subject to sampling error. Hence, we balance the precision of smaller buckets with the increased uncertainty when choosing parameters.

To begin, we first consider the accuracy in our estimation using a score histogram (without privacy noise) with $O(B)$ buckets. If the value of the calibrated score function varies arbitrarily as the uncalibrated score changes, then calibration via histogram is not a meaningful task. Hence, we assume the $(\phi, O(1))$-well-behaved property of the (ideal) calibrated score function (Defn 1), meaning that each bucket is either heavy or smoothly varying.

Theorem 7. *In the Federated case, the expected calibration error using score histograms is bounded by* $O(1/M^{1/3})$.

Proof. Recall that the (ideal) calibrated value for a score s is the true positive ratio at that point, i.e., $p(s)/(p(s) + n(s))$, where $p(s)$ and $n(s)$ are the probability mass functions for positive and negative examples. We assume that this calibration function $c(s)$ is $(1/B, \ell)$-well behaved, for a constant ℓ, so that between any spikes the maximum change is governed by $|c(s)-c(s+\Delta)| \le \ell\Delta$. We consider the behavior of the score function within a histogram bucket of width Δ that includes the score value s. So the calibrated value for any point in the bucket must be in the range $c(s) \pm \ell\Delta = c(s) \pm \ell/B$, as we require the width of any histogram bucket to be at most $1/B$.

Table 2: Data and classifiers from three different "tabular playground" Kaggle competitions used for evaluation.

Kaggle challenge	Classification task domain	Data rows	Baseline classifier	ROC AUC
Sep 2021	Insurance risk	958K	LightGBM	0.79
Oct 2021	Genetic tests	1M	XGBoost	0.85
Nov 2021	Email spam	600K	Logit Regression	0.73

The points drawn in the histogram for this bucket can be considered to be samples, where the probability of each sample for score s' being a positive example is $c(s')$. By a standard Hoeffding bound, the probability that the mean calibrated value of n sampled points falls below $c(s) - \ell/B - \varepsilon$, or exceeds $c(s) + \ell/B + \varepsilon$ is bounded by $\exp(-2\varepsilon^2 n)$. Since each bucket has $\Theta(M/B)$ samples, we set this probability to be a small constant and rearrange to guarantee that for any score s' that falls in the bucket we can estimate its calibrated value within absolute error at most $\ell/B + \sqrt{B/M}$. Else, if the bucket contains a spike, then the error is dominated by the sampling error, and so we focus on the non-spiky case. Trading off these two error terms, we equate $\ell/B = \sqrt{B/M}$. Rearranging, and treating ℓ as a constant, we set $B \propto M^{1/3}$ to balance these errors. In this case, the (expected) error achieved is $O(M^{-1/3})$. □

Hence, calibration is challenging, as accuracy improves slowly as a function of the number of clients, M (due to the uncertainty from client sampling). Next, we consider the impact of privacy noise.

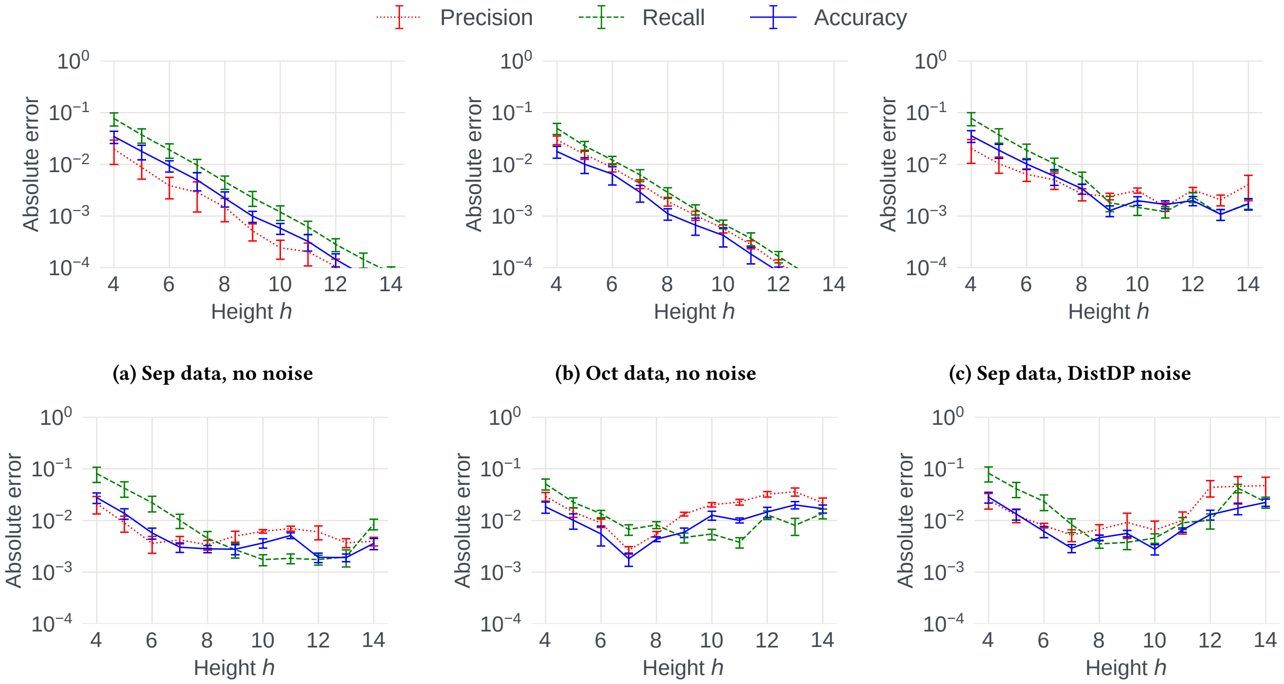
Theorem 8. *Expected calibration error in the LocalDP and DistDP cases is* $O(1/\epsilon^{1/2}M^{1/4})$ *and* $O(1/M^{1/3} + 1/\epsilon M^{2/3})$*, respectively.*

Proof. We follow the argument of Theorem 7 to argue that within a bucket we have $\Theta(M/B)$ points. However, now the estimate from the bucket is perturbed due to privacy noise. In particular, we obtain a value for p_i and n_i, the numbers of positive and negative examples in the bucket, that have expected absolute error of $\sqrt{M}/\epsilon$ (in the LocalDP case) or $1/\epsilon$ (in the DistDP case). In a bucket with $\Theta(M/B)$ examples, this yields an additional error on estimates of c of $O(B/\epsilon\sqrt{M})$ or $O(B/\epsilon M)$, respectively.

For the LocalDP case, the quantity of $B/\epsilon\sqrt{M}$ will dominate the $\sqrt{B/M}$ term, leading us to choose $B = O(\sqrt{\epsilon M^{1/2}})$. This sets the error bound to $O(\epsilon^{-1/2}M^{-1/4})$. That is, we expect the number of bins needed to be rather small in the LDP case.

For the DistDP case, the quantity of $B/\epsilon M$ will be of lower magnitude than $\sqrt{B/M}$ since (treating ϵ as a constant) $B/M < 1$. Hence, we focus on balancing ℓ/B with $\sqrt{B/M}$ as in the noiseless case. This sets $B = O(M^{1/3})$ to achieve error $O(M^{-1/3} + 1/\epsilon M^{2/3})$. We state this as $O(M^{-1/3})$, assuming that $1/M^{1/3} < \epsilon$. □

For DistDP, we expect that there are enough clients so that $1/M^{1/3} < \epsilon$, thus the bound simplifies to $O(M^{-1/3})$. The dependence on ϵ is (surprisingly) limited because sampling noise dominates privacy noise. So we anticipate small difference in calibration quality with and without privacy noise. For LocalDP, privacy noise is big enough to affect the error but only weakly (as $1/\sqrt{\epsilon}$).



(a) Sep data, no noise

(b) Oct data, no noise

(c) Sep data, DistDP noise

(d) Nov data, DistDP noise

(e) Oct data, LocalDP noise

(f) Nov data, LocalDP noise

Figure 2: Accuracy for classifier Precision, Recall and Accuracy estimation with varying noise levels

7 EMPIRICAL EVALUATION

Our empirical study quantifies the accuracy of the proposed approaches and the impact of enforcing different models of privacy. We simulate a distributed environment on a single CPU and evaluate several approaches for a selection of trained classifiers using data examples with ground-truth labels and predicted scores [4]. We leverage freely-available synthetic data and trained baseline classifiers from three diverse “tabular playground” competitions at Kaggle.[5] These data science challenges present a variety of realistic synthetic data and prediction tasks. Table 2 summarizes the different data and classifiers used from the three different challenges from 2021. They each have binary targets and approximately balanced positive and negative classes. The input is then the set of scored examples and labels, i.e., the pairs $(w(x_i), y_i)$, where each client is assigned one example. We show representative results on subsets of the datasets; full results are in our extended technical report [6].

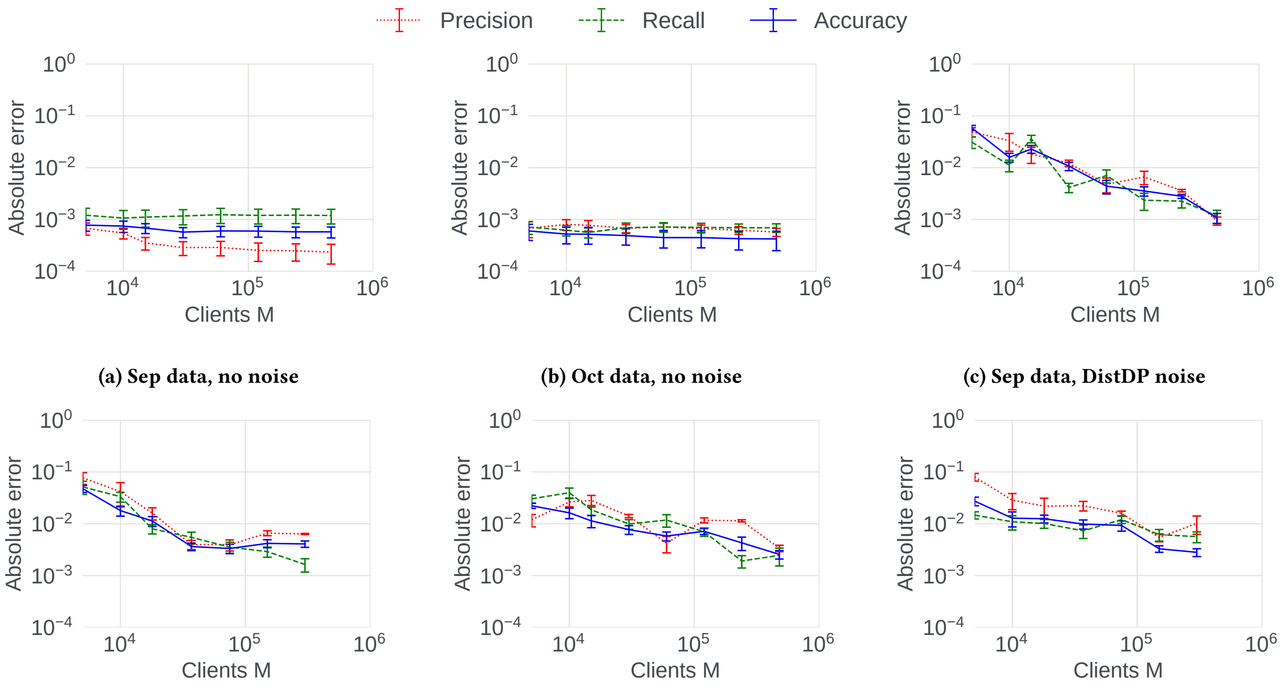
Privacy settings. Choosing privacy parameter ϵ is the subject of several studies [15, 17]. Across many implementations [7, 10], $\epsilon < 1$ is considered very strong privacy, while $\epsilon > 10$ is very weak, and typical choices fall between these. We evaluate our results with ϵ values between these extremes, using standard noise distributions. For LocalDP, we implement Optimized Unary Encoding [28] with $\epsilon = 5.0$, a common setting, which was slightly preferable to other LocalDP mechanisms. For DistDP, we sample discrete Laplace noise via the summation of M Pólya distributions, equivalent to $\epsilon = 1.0$.

7.1 Precision, Recall, and Accuracy

Figure 2 shows results for estimating precision, recall and accuracy as we vary the parameter h that determines the height of the hierarchy used for the score histogram. For each data set, we consider ten different decision score thresholds T $(1/11, 2/11 \ldots 10/11)$ to define a binary classifier. For each experiment, we show the absolute error between the exact and reported values, with error bars showing the variation over the threshold choices. Figures 2a-2b show that in the Federated setting the error behavior is similar across datasets. Error decreases rapidly as the h increases, as this gives more fidelity to represent the score distribution, in accordance with Theorem 2. Of the three metrics, *precision* has slightly higher error, consistent with the proof of Theorem 2: we approximate both terms in the ratio, whereas for recall and accuracy, we only approximate the numerator of the ratio. The total error can be made arbitrarily small, e.g., $< 10^{-4}$ for $h = 14$, sufficient to compare two classifiers accurately.

With DistDP noise (Figures 2c-2d) there is now a tradeoff between (i) better data descriptions with a taller hierarchy, and (ii) the extra privacy noise due to more numerous buckets. The results are consistent with Theorem 2, which predicts that h should be chosen close to 12. Indeed, we observe the lowest error near this predicted value, around $h = 11$ for Sep, and $h = 10$ for Nov, yielding error around 0.001. For LocalDP noise (Figures 2e-2f), the tradeoff is shallower, and the lowest error is seen around 0.005. Theorem 2 suggests choosing the number of buckets proportional to $\epsilon^{2/3}M^{1/3}$, which means $h = 8$ for these experiments and indeed marks the lowest error for estimating precision, recall and accuracy.

[4] Our Python notebook is available at https://figshare.com/s/607998e479b0778645f6
[5] See https://www.kaggle.com/competitions?hostSegmentIdFilter=8 Accessed: 7/7/23



(a) Sep data, no noise

(b) Oct data, no noise

(c) Sep data, DistDP noise

(d) Nov data, DistDP noise

(e) Oct data, LocalDP noise

(f) Nov data, LocalDP noise

Figure 3: Accuracy for classifier Precision, Recall and Accuracy estimation with varying population size

Figure 3 varies the number of clients, M, in the three privacy regimes (with $h = 10$ based on earlier experiments). Now we see no observable impact in the Federated case (Figures 3a to 3b): accuracy is not affected by increasing M. For DistDP noise (Figures 3c to 3d) and LocalDP noise (Figures 3e to 3f), errors drop as M increases, consistent with the $O(1/M)$ behavior seen in Theorem 2. So, increasing M by 10× reduces error by this factor. Good accuracy with DP noise requires a population size $>$ 10K, and $>$ 100K for LocalDP.

7.2 Area Under Curve

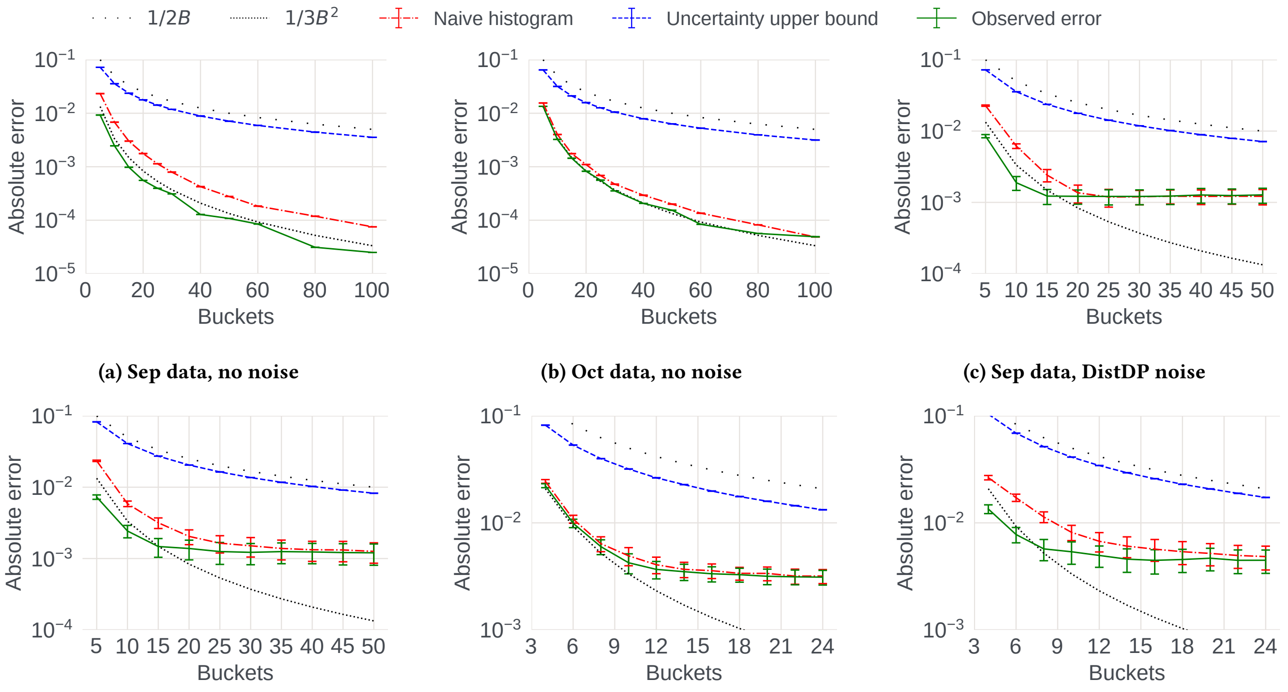
For Area Under Curve (AUC), we show the results over ten repetitions, varying (*i*) how the examples are sampled, and (*ii*) the random noise. For all methods, we use a hierarchy with $h = 10$, found earlier to be a good choice. Figure 4 shows our results for AUC, as we vary the data and the noise model. In each plot, the guideline $1/2B$ represents the pessimistic bound from Lemma 3, while the guideline $1/3B^2$ shows our tighter bound under the *well-behaved* assumption (Theorem 4). For each experiment, we plot a line showing the worst-case uncertainty in our estimate, due to the noise in each bucket. That is, the quantity corresponding to $\sum_i p_i n_i$, the sum over buckets of the product of the number of positive and negative examples. This is the error we would see if the analysis in Lemma 3 was tight. We also plot one curve for using the histogram naively, i.e., picking B buckets with uniform boundaries, and the observed error for our approach where we pick B buckets based on the (estimated) quantile boundaries. This uniform choice of buckets is equivalent to the recent approach in the LocalDP model [27]: as we will see, it is outperformed by the quantile histogram approach.

In the Federated case (no explicit privacy noise), the worst-case error bound indeed follows $O(1/B)$, but our tighter analysis yields $O(1/B^2)$ errors : the total uncertainty follows the $1/2B$ curve closely, while the histogram estimators follow the $1/3B^2$ curve. The error vanishes: with 100 buckets, AUC is estimated with 10^{-5} accuracy, sufficient for all uses. Quantile-based histograms produce smaller errors than uniform-bucket boundaries, by an order of magnitude.

Theorem 5 predicts an accuracy limit due to a fixed level of noise from DistDP privacy. Experiments confirm this: the error curve initially follows $1/3B^2$ but then flattens after about 20-40 buckets. Here, the AUC estimation error is ≈0.001—small enough for useful conclusions about the classifier. With more buckets, examples distribute across them without large clusters, helping the uniform and quantile-based histograms work as well. The same behavior holds for the LocalDP case, where the error bound converges to ≈ 0.005. The speed of convergence and value reached vary based on the data used. Beyond 20 buckets, error reduction is minimal, as LocalDP noise has stronger impact than the DistDP noise.

7.3 Calibration

Recall that expected calibration error (ECE) is found by dividing the domain of the scores into bins (distinct from the histogram buckets used in our algorithms) and measuring the fraction of positive examples in each bin, averaged over all bins. Plots in Figure 5 vary the number of such bins and allow our calibration approach to use the same number of buckets as bins. We adapt the Bayesian Binning into Quantiles (BBQ) technique of Naeini et al. [22], which also makes use of histograms based on quantiles, by trying a range



(a) Sep data, no noise

(b) Oct data, no noise

(c) Sep data, DistDP noise

(d) Nov data, DistDP noise

(e) Oct data, LocalDP noise

(f) Nov data, LocalDP noise

Figure 4: Accuracy for ROC AUC estimation with varying noise levels

of choices of B from $M^{1/3}/10$ up to $10M^{1/3}$. Each choice of B is assigned a BBQ score based on the number of positive and negative examples in each bucket, via the Gamma function, which is used to compute a weighted sum of binning choices. We can implement this approach in our setting, as high-resolution histograms can be used to build the score histograms for many choices of B. In the Federated case, this should give the same results as BBQ in the centralized setting. For histograms with privacy noise, we expect some deviation in performance, since the BBQ method is not tuned to correct for the noise in bucket counts. On these plots, we show the calibration error identified by the Bayesian Binning into Quantiles (BBQ) method combined with our histogram approach. The baselines are (*i*) the calibration error of the uncalibrated score function and (*ii*) the result of using the (centralized) implementation of isotonic regression from `scikit-learn 0.22`.

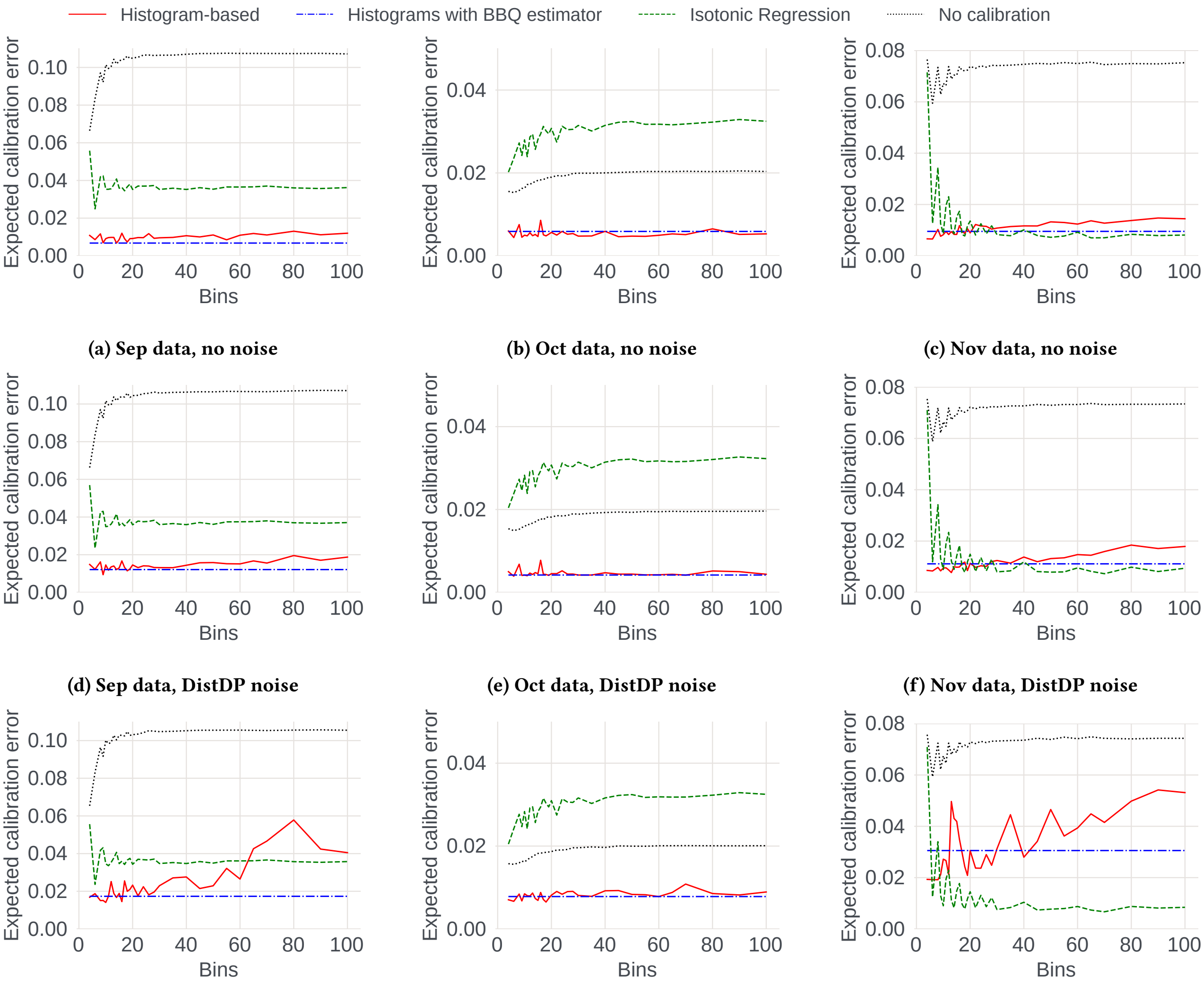
For the Federated case, plots in Figures 5a-5c show that good accuracy is possible – calibration error of ≈0.01 is achievable, i.e., on average, the calibrated score is within 0.01 of the true probability. This outcome is not very sensitive to the number of evaluation bins. The BBQ approach on top of our histogram approach does a good job at combining information from multiple bucketings when there is no noise, and gives a reliable choice of calibration. We note that, due to the use of the Gamma function in defining scores, it often happens that one bucketing has a vastly greater BBQ score than other choices. Then normalized weighting puts all weight on this bucketing, so the method effectively simplifies into choosing the number of buckets. This gives an improvement over using the uncalibrated score function, where the calibration error can be much larger, ≈0.1 for Sep and 0.08 for Nov. Surprisingly, the (centralized) isotonic approach is not a good fit for these score functions. On Sep data, it attains calibration error of 0.04, and for Oct data it increases the error compared to the original score function. Isotonic regression only clearly helps for the Nov data.

Introducing DistDP noise does not change the results much, as anticipated by our observation in Theorem 8 that privacy noise is outweighed by the variation of data points within the bins. Further, the overall calibration error is similar in magnitude to the Federated case, ≈0.01. For LocalDP noise, the error increases to 0.02 and higher, as the impact of privacy noise is noticeable. Given the choice of the number of buckets, using fewer calibration buckets reduces noise. Despite cruder calibration, using ≤10 buckets keeps the error near 0.02. As expected, the BBQ approach is affected by the extra noise, and tends to place more weight on choices with more buckets. For Oct data, the original uncalibrated score function has smaller error, and combining LocalDP noise with calibration causes more harm than good. For Sep and Nov, where the original score function was not well calibrated, federated calibration brings significant benefit.

7.4 Other Experimental Observations

Dependence on privacy parameter ϵ. We found that the bounds in Table 1 were closely followed in our experiments as we varied ϵ. This is unsurprising, since the impact of varying ϵ for histograms is well-understood, and the impact on accuracy is quite direct.

Time cost. Simulations were performed on a single CPU machine, and were not highly optimized for performance. Nevertheless, we accurately simulated the tasks of each client and the server within



(a) Sep data, no noise

(b) Oct data, no noise

(c) Nov data, no noise

(d) Sep data, DistDP noise

(e) Oct data, DistDP noise

(f) Nov data, DistDP noise

(g) Sep data, LocalDP noise

(h) Oct data, LocalDP noise

(i) Nov data, LocalDP noise

Figure 5: Classifier calibration accuracy with varying noise levels

the protocol. Typically, it took only minutes to evaluate a large range of parameter choices and repetitions, thus the cost per client is trivial (milliseconds of computation effort per client), and the effort for the server is just simple aggregation.

Dependence on Number of Clients. For AUC and calibration, we saw that accuracy improves as M increases, more quickly for DistDP and LocalDP. For Federated AUC, there is no impact of increasing M, while for calibration, the ECE for DistDP and Federated converges.

8 CONCLUDING REMARKS

Distributed data management system support for federated learning requires federated calibration and computation of classifier metrics in order to maintain end-to-end privacy. Our results demonstrate feasibility for these key tasks. Many other aspects of distributed data management also need federated solutions: data cleaning, feature selection, and normalization, etc. We expect approaches similar to our histogram-based algorithms will apply here. Importantly, the use of histograms is robust to the heterogeneity that is rampant in the distributed setting. The end goal for this line of work will be to build systems that achieve end-to-end privacy guarantees for federated learning, from feature extraction to deployment with ongoing performance tracking, and so on. Extending to entire distributed systems would require some consideration of *privacy budgeting* across tasks to support a single ϵ-DP end-to-end guarantee. The challenge is to determine how best to divide ϵ among different stages and ensure sufficient performance.

ACKNOWLEDGMENTS

We thank Akash Bharadwaj, Ilya Mironov, Peter Romov, Harish Srinivas, Daniel Ting and the anonymous reviewers for helpful comments and feedback.

REFERENCES

[1] Naman Agarwal, Peter Kairouz, and Ziyu Liu. 2021. The Skellam Mechanism for Differentially Private Federated Learning. *CoRR* abs/2110.04995 (2021), 25. arXiv:2110.04995 https://arxiv.org/abs/2110.04995

[2] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2020. Private Summation in the Multi-Message Shuffle Model. In *ACM SIGSAC Conference on Computer and Communications Security*. ACM, 657–676. https://doi.org/10.1145/3372297.3417242

[3] James Bell, Aurélien Bellet, Adrià Gascón, and Tejas Kulkarni. 2020. Private Protocols for U-Statistics in the Local Model and Beyond. In *Int'l Conf. Artificial Intelligence and Statistics, AISTATS (Proc. Machine Learning Research)*, Vol. 108. PMLR, 1573–1583. http://proceedings.mlr.press/v108/bell20a.html

[4] Aloni Cohen and Kobbi Nissim. 2020. Towards formalizing the GDPR's notion of singling out. *Proc. Natl. Acad. Sci. USA* 117, 15 (2020), 8344–8352. https://doi.org/10.1073/pnas.1914598117

[5] Graham Cormode, Tejas Kulkarni, and Divesh Srivastava. 2019. Answering Range Queries Under Local Differential Privacy. *Proc. VLDB Endow.* 12, 10 (2019), 1126–1138. https://doi.org/10.14778/3339490.3339496

[6] Graham Cormode and Igor L. Markov. 2022. Federated Calibration and Evaluation of Binary Classifiers. *CoRR* abs/2210.12526 (2022), 24. https://doi.org/10.48550/arXiv.2210.12526 arXiv:2210.12526

[7] Damien Desfontaines. 2021. A list of real-world uses of differential privacy. https://desfontain.es/privacy/real-world-differential-privacy.html.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conf. North American Chapter of the ACL: Human Language Technologies, NAACL-HLT*. Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[9] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT (Lecture Notes in Computer Science)*, Vol. 4004. Springer, 486–503. https://doi.org/10.1007/11761679_29

[10] Cynthia Dwork, Nitin Kohli, and Deirdre K. Mulligan. 2019. Differential Privacy in Practice: Expose your Epsilons! *J. Priv. Confidentiality* 9, 2 (2019), 22. https://doi.org/10.29012/jpc.689

[11] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3-4 (2014), 211–407. http://dblp.uni-trier.de/db/journals/fttcs/fttcs9.html#DworkR14

[12] Marco Gaboardi, Ryan Rogers, and Or Sheffet. 2019. Locally Private Mean Estimation: Z-test and Tight Confidence Intervals. In *Int'l Conf. Artificial Intelligence and Statistics, AISTATS (Proc. Machine Learning Research)*, Vol. 89. PMLR, 2545–2554. http://proceedings.mlr.press/v89/gaboardi19a.html

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proc. 34th International Conference on Machine Learning (Proc. Machine Learning Research)*, Vol. 70. PMLR, 1321–1330. http://proceedings.mlr.press/v70/guo17a.html

[14] David J. Hand and Robert J. Till. 2001. A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Mach. Learn.* 45, 2 (2001), 171–186. https://doi.org/10.1023/A:1010920819831

[15] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C. Pierce, and Aaron Roth. 2014. Differential Privacy: An Economic Method for Choosing Epsilon. In *IEEE Computer Security Foundations Symposium*. IEEE Computer Society, 398–410. https://doi.org/10.1109/CSF.2014.35

[16] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista A. Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. 2021. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* 14, 1-2 (2021), 1–210. https://doi.org/10.1561/2200000083

[17] Jaewoo Lee and Chris Clifton. 2011. How Much Is Enough? Choosing ϵ for Differential Privacy. In *Information Security (Lecture Notes in Computer Science)*, Vol. 7001. Springer, 325–340. https://doi.org/10.1007/978-3-642-24861-0_22

[18] Gregory J Matthews and Ofer Harel. 2013. An examination of data confidentiality and disclosure issues related to publication of empirical ROC curves. *Academic radiology* 20, 7 (July 2013), 889–896. https://doi.org/10.1016/j.acra.2013.04.011

[19] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Int'l Conf. Artificial Intelligence and Statistics, AISTATS (Proc. Machine Learning Research)*, Vol. 54. PMLR, 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html

[20] Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the Calibration of Modern Neural Networks. *CoRR* abs/2106.07998 (2021), 28. arXiv:2106.07998 https://arxiv.org/abs/2106.07998

[21] Rajeev Motwani and Prabhakar Raghavan. 1995. *Randomized Algorithms*. Cambridge University Press.

[22] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. In *AAAI Conf. Artificial Intelligence*. AAAI Press, 2901–2907. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9667

[23] Stuart J. Russell and Peter Norvig. 2020. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson. http://aima.cs.berkeley.edu/

[24] Aaron Segal, Antonio Marcedone, Benjamin Kreuter, Daniel Ramage, H. Brendan McMahan, Karn Seth, K. A. Bonawitz, Sarvar Patel, and Vladimir Ivanov. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *CCS*. ACM, 21. https://eprint.iacr.org/2017/281.pdf

[25] Ben Stoddard, Yan Chen, and Ashwin Machanavajjhala. 2014. Differentially Private Algorithms for Empirical Machine Learning. *CoRR* abs/1411.5428 (2014), 13. arXiv:1411.5428 http://arxiv.org/abs/1411.5428

[26] Jiankai Sun, Xin Yang, Yuanshun Yao, Junyuan Xie, Di Wu, and Chong Wang. 2022. Differentially Private AUC Computation in Vertical Federated Learning. https://doi.org/10.48550/ARXIV.2205.12412

[27] Jiankai Sun, Xin Yang, Yuanshun Yao, Junyuan Xie, Di Wu, and Chong Wang. 2022. DPAUC: Differentially Private AUC Computation in Federated Learning. https://doi.org/10.48550/ARXIV.2208.12294

[28] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium, USENIX Security*. USENIX Association, 729–745. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/wang-tianhao

[29] S. L. Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69.

[30] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi, Shi Jin, Tony Q. S. Quek, and H. Vincent Poor. 2019. Federated Learning with Differential Privacy: Algorithms and Performance Analysis. *CoRR* abs/1911.00222 (2019), 15. arXiv:1911.00222 http://arxiv.org/abs/1911.00222

[31] Jianyu Yang, Tianhao Wang, Ninghui Li, Xiang Cheng, and Sen Su. 2020. Answering Multi-Dimensional Range Queries under Local Differential Privacy. *Proc. VLDB Endow.* 14, 3 (2020), 378–390. https://doi.org/10.5555/3430915.3442436

[32] Mengmeng Yang, Lingjuan Lyu, Jun Zhao, Tianqing Zhu, and Kwok-Yan Lam. 2020. Local Differential Privacy and Its Applications: A Comprehensive Survey. *CoRR* abs/2008.03686 (2020), 25. arXiv:2008.03686 https://arxiv.org/abs/2008.03686

[33] Lingchen Zhao, Jianlin Jiang, Bo Feng, Qian Wang, Chao Shen, and Qi Li. 2021. SEAR: Secure and Efficient Aggregation for Byzantine-Robust Federated Learning. *IEEE Trans. on Dependable and Secure Computing* 19, 5 (2021), 3329 – 3342. https://doi.org/10.1109/TDSC.2021.3093711