Quality Inference in Federated Learning with Secure Aggregation

Balázs Pejó and Gergely Biczók

CrySyS Lab, Department of Networked Systems and Services, Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics
{pejo,biczok}@crysys.hu

Abstract

Federated learning algorithms are developed both for efficiency reasons and to ensure the privacy and confidentiality of personal and business data, respectively. Despite no data being shared explicitly, recent studies showed that it could still leak sensitive information. Hence, to prevent attribution to specific participants secure aggregation is utilized in many real-world scenarios. In this paper, we focus on the quality of individual training datasets and show that such information could be inferred and attributed to specific participants even when secure aggregation is applied.

More precisely, through a series of image recognition experiments, we infer the relative quality ordering of participants. Moreover, in two example use cases, we apply the inferred quality information to stabilize the training performance and to detect misbehaviours.

Introduction

For Machine Learning tasks, it is widely accepted that more training data leads to a more accurate model. Unfortunately, in reality, the data is scattered among multiple different entities. Thus, data holders could potentially increase the accuracy of their local model accuracy by training a joint model together with others (Pejo, Tang, and Biczok 2019). Several collaborative learning approaches were proposed in the literature, amongst which the least privacy friendly method is centralized learning, where a server pools the data from all participants together and trains the desired model. On the other end of the privacy spectrum, there are cryptographic techniques such as multi-party computation (Goldreich 1998) and homomorphic encryption (Gentry et al. 2009), guaranteeing that only the final model is revealed to legitimate collaborators and nothing more. Neither of these extremes admit most of the real-world use-cases: while the first requires participants to share their datasets directly, the latter requires too much computational resource to be a practical solution.

Somewhere between these (in terms of privacy protection) stands federated learning (FL) which mitigates the communication bottleneck and provides flexible participation by selecting a random subset of participants per round, who compute and send their model updates to the aggregator server (Konečný et al. 2016). FL provides some privacy protection by design as the actual data never leaves the hardware located within the participants' premises[1]. Yet, there is an already rich and growing related literature revealing that from these updates (i.e., gradients) a handful of characteristics can be inferred about the underlying training dataset. For instance, source inference attack could tie the extracted information to specific participants of FL (Hu et al. 2021). Parallel to these, several techniques have been developed to conceal the participants' updates from the aggregator server, such as differential privacy (Desfontaines and Pejó 2020) and secure aggregation (*SA*) (McMahan et al. 2016). Although the first approach comes with a mathematical privacy guarantee, it also results in heavy utility loss, which limits its applicability in many real-world scenarios. On the other hand, *SA* does not affect the aggregated final model, which makes it a suitable candidate for many applications. Essentially, *SA* hides the individual model updates without changing the aggregated model by adding pairwise noise to the participants' gradients in a clever way so that they cancel out during aggregation.

Consequently, *SA* only protects the participants' individual updates, and leaves the aggregated model unprotected. *SA* provides a "hiding in the crowd" type of protection, thus, without specific background knowledge, it is unlikely that an attacker could link the leaked information to a specific participant. In this paper we study the possibility of *inferring the quality of the individual datasets when* SA *is in place*. Note that quality inference is different from poisoning attack detection (Bagdasaryan et al. 2020), as that line of research is only interested in classifying participants as malicious or benign, while our goal is to enable the fine-grained differentiation of the participants with respect to input quality. To the best of our knowledge we are the first to study this problem in a *secure aggregation* setting.

Contributions. Our method recovers the quality of the aggregated updates[2]; consequently, the quality of the contribut-

[1] Or, in case of cloud computing, the compute/storage instance controlled by the given participant

[2] Such a quality measure is relative to the particular task and to the other participants' datasets. Therefore, we aim to retrieve a relative quality ordering of the participants (i.e., compared to each other for the particular task).

ing participants' datasets. To obtain this quality information, our method takes advantage of the inferred information across the aggregated updates of multiple rounds and the known per-round subset of participants associated with the corresponding aggregates. We recover the relative quality ordering by evaluating the aggregated updates in each round, and assigning scores to the contributors based on three simple rules called *The Good*, *The Bad*, and *The Ugly* (IMDB 1966).

We conduct experiments on two neural network architectures (MLP and CNN) on two datasets (MNIST and CIFAR10) with three settings (2 our of 5, 5 out of 25, and 10 out of 100 participants are selected in each round to update the model). Our experiments show that the three proposed simple heuristic scoring rules significantly outperform the baseline in ordering the participants based on their data quality. Moreover, we find that the accuracy of quality inference depends on both the complexity of the task and the trained model architecture.

We also experiment with small adjustments to the proposed rules to fine-tune their hyperparameters. Finally, we investigate two potential applications of quality inference: on-the-fly performance boosting and misbehaviour detection. We find that i) carefully weighting the participants based on the inferred quality smooths the learning curve as well as improves the trained model's accuracy and ii) the scores can be used to detect both malicious misbehavior and free-riding. We are not aware of any work tackling any of the aforementioned issues when *SA* is enabled.

The Theoretic Model

In this section we introduce the theoretical model of quality inference and highlight its complexity. We note with n a participant in FL, while N denotes the number of all participants. Similarly, i denotes a round in FL, while I denotes the number of all rounds. The set S_i contains the randomly selected participants for round i, and $b = |S_i|$ captures the number of selected participants. D_n is participant n's dataset consisting of $(x, y) \in D_n$ data-label pairs. We assume D_n is associated with a single scalar u_n, which measures its quality. We use θ_n and v_i to capture the quality of the nth participant's gradient and the quality of the aggregated gradient in the ith round, respectively. A summary of the variables are listed in Table 1.

Variable	Description
$n \in [N]$	Participants
$i \in [I]$	Training rounds
S_i	Set of selected par. for round i
b	Number of selected participants
$(x, y) \in D_n$	Dataset of participant n
u_n	Quality of D_n
v_i	Quality of aggr. gradient in round i
θ_n	Quality of participant n's gradient

Table 1: Notation used in the paper.

Deterministic Case. In this simplified scenario we assume the gradient quality is equal to the dataset quality, i.e., $\theta_n = u_n$. Consequently, the aggregated gradients represent the average quality of the participants' datasets. As a result, the round-wise quality values of aggregated gradients form a linear equation system $Au = v$, where $u = [u_1, \ldots, u_N]^T$, $v = [v_1, \ldots, v_I]^T$, and $a_{i,n} \in A_{I \times N}$ indicates whether participant n is selected for round i. Depending on the dimensions of A, the system can be under- or over-determined. In case of $I < N$ (i.e., no exact solution exists) and if $I > N$ (i.e., many exact solutions exist), the problem itself and the approximate solution are shown in Eq. 1 and 2, respectively.

$$\min_u ||v - Au||_2^2 \quad \Rightarrow \quad u = (A^T A)^{-1} A^T v \tag{1}$$

$$\min_u ||u||_2^2 \text{ s.t. } Au = v \quad \Rightarrow \quad u = A^T (AA^T)^{-1} v \tag{2}$$

Stochastic Case. The above equations do not take into account any randomness. Given that the training is stochastic, we can treat the quality of participant n's gradient as a random variable θ_n sampled from a distribution with parameter u_n. Moreover, we can represent $\theta_n = u_n + e_n$ where e_n corresponds to a random variable sampled from a distribution with zero mean. We can further assume that e_n and $e_{n'}$ are i.i.d. for $n \neq n'$. As a result, we can express the aggregated gradient $v_i = \sum_n a_{i,n} u_n + E$ where E is sampled from the convolution of the probability density function of e's.

In this case, due to the Gauss–Markov theorem (Harville 1976), the solution in Eq. 1 is the best linear unbiased estimator, with error $||v - Au||_2^2 = v^T(\mathbf{I} - A(A^T A)^{-1} A^T)v$ (where $\mathbf{I}$ is the identity matrix) with an expected value of $b(\mathbf{I} - N)$. Note, that with more iterations more information is leaking, which should decrease the error. Yet, this is not captured by the theorem as it considers every round as a new constraint.

This problem lies within estimation theory (Ludeman 2003), from which we already know that estimating a single random variable with added noise is already hard; moreso, factoring in that in our setting, we have multiple variables forming an equation system. Moreover, these random variables are different per round; a detail we have omitted thus far. Nevertheless, each iteration corresponds to a different expected accuracy improvement level, as with time the iterations improve less-and-less. Consequently, to estimate individual dataset quality we have to know the baseline expected learning curve; in turn, the learning curve depends exactly on those quality values. Being a chicken-egg problem, we focus on empirical observations to break this vicious cycle.

Quality Scoring

In this section we devise three intuitive scoring rules that either rewards or punishes the participants in the FL rounds. We summarize our notations in Table 2. Note that in the rest of the paper we slightly abuse the notation φ and q by removing index i where it is not relevant.

Assumptions. We assume a honest-but-curious setting; the aggregator server (and the participants) can only observe passively. Further restrictions on the attacker include limited

Variable	Description
ω_i	Model improvement in the ith round
$\varphi_{i,n}$	Quality score of par. n after round i
$q_{i,n}$	Inferred quality-wise rank of participant n after round i
d_s	Spearman Distance
r_s	Spearman Coefficient

Table 2: Notation for the scoring rules.

computational power and no background knowledge besides access to an evaluation oracle. For this reason, we neither utilize any contribution score based techniques nor existing inference attacks, as these require either significant computational resources or user-specific relevant background information.

Scoring Rules. Based on the round-wise improvements ω_i, we created three simple rules to reward or punish the participants. We named them *The Good*, *The Bad*, and *The Ugly*; the first one rewards the participants in the more useful aggregates, the second punishes in the less useful ones, while the last one punishes when the aggregate does not improve the model at all.

- *The Good*: each participant contributing in round i which improves the model more than the previous round (i.e., $\omega_i > \omega_{i-1}$) receives $+1$.
- *The Bad*: each participant contributing in round i which improves the model less than the following round (i.e., $\omega_i < \omega_{i+1}$) receives -1.
- *The Ugly*: each participant contributing in round i which does not improve the model at all (i.e., $\omega_i < 0$) receives -1.

It is reasonable to expect that the improvements in consecutive rounds are decreasing (i.e., $\omega_i < \omega_{i-1}$): first the model improves rapidly, while improvement slows down considerably in later rounds. The first two scoring rules (*The Good* and *The Bad*) capture the deviation from this pattern: we can postulate that i) high dataset quality increases the improvement more than in the previous round, and ii) low dataset quality decreases the improvement, which would be compensated in the following round. These phenomena were also shown in (Kerkouche, Ács, and Castelluccia 2020). Our last scoring rule (*The Ugly*) is built on the premise that if a particular round does not improve the model, there is a higher chance that some of the corresponding participants have supplied low quality data.

Independently of the participants' dataset qualities, round-wise improvements could deviate from this pattern owing to the stochastic nature of learning. We postulate that this affects all participants evenly, independently of their dataset quality; thus, the relation/ordering among the individual scores are not significantly affected by this "noise". Participant selection also introduces a similar round-wise "noise"; however, we assume that participants are selected uniformly, hence, its effect should also be similar as per participant.

Quantifying Quality Inference. To quantify the inferred quality ordering of the participants, we need to convert the relation between the quality scores into a single value. For this purpose, we use the Spearman correlation coefficient r_s (Zar 2005), which is based on the Spearman distance d_s (Diaconis and Graham 1977). By accumulating the quality scores of the participant after every iteration we can establish the current quality-wise ordering. For instance, $q_{i,n} = 0$ means $\varphi_{i,n} \leq \varphi_{i,n'}$ for all $n' \in [0, N]$, i.e., participant n has the lowest score after iteration i. The Spearman distance measures the absolute difference of this inferred and the actual position. The Spearman correlation coefficient assesses monotonic relationships on the scale $[-1, 1]$; 1 corresponds to perfect correlation (i.e., perfect quality-wise ordering), while any positive value signals positive correlation between the actual and the inferred quality ordering.[3] Note, that the Spearman distance (and consequently the coefficient) handles any misalignment equally irrespective of the position; these are calculated according to the Eq. 3.

$$d_s(i,n) = |n - q_{i,n}| \quad r_s(i) = 1 - \frac{6 \cdot \sum_{n=1}^{N} d_s(i,n)^2}{N \cdot (N^2 - 1)} \quad (3)$$

Experiments for Quality Inference

In this section we describe our experiments in detail, including the evaluation and two potential applications of quality inference.

Simulating Data Quality. Data quality in general is relative for two reasons: it can only be considered in terms of the proposed use, and in relation to other examples. Data quality entails multiple aspects such as accuracy, completeness, redundancy, readability, accessibility, consistency, usefulness, and trust, with several having their own subcategories (Batini, Scannapieco et al. 2016). We focus on image recognition tasks as it is a key ML task with standard datasets available. Still, we have to consider several of these aspects in relation with image data. Visual perception is a complex process; to avoid serious pitfalls, we do not manipulate the images themselves to simulate different qualities. Rather, as we focus on supervised machine learning, we modify the label y corresponding to a specific image x. To have a clear quality-wise ordering between the datasets, we perturbed the labels of the participants according to Eq. 4, where ψ_k is drawn uniformly at random over all available labels. Putting it differently, the labels of the participants' datasets are randomized before training with a linearly decreasing probability from 1 to 0 based on their IDs.[4]

$$\Pr(y_k = \psi_k | (x_k, y_k) \in D_n) = \frac{N - n}{N - 1} \quad (4)$$

[3] E.g., if the rules determine 5-3-2-4-1 as quality ordering while the actual order is 5-4-3-2-1, then the Spearman distances are 0-2-1-1-0, and the Spearman correlation is 0.7, suggesting that the inferred quality order is very close to the original one.

[4] No bias is introduced as both the initial dataset splitting and the round-wise participant selection are random.

Algorithm 1: Quality Inference in FL w/ SA

Input: data D; participants N; rounds I

1: $Split(D, N) \rightarrow \{D_1, \ldots, D_N, D_{N+1}\}$
2: **for** $n \in [1, \ldots, N]$ **do**
3: $\quad \forall\ (x_k, y_k) \in D_n : y_k \sim$ Eq. 4
4: $\varphi = [0, \ldots, 0]$; $M_0 \leftarrow Rand()$
5: **for** $i \in [1, \ldots, I]$ **do**
6: $\quad RandSelect([1, \ldots, N], b) \rightarrow S_i$
7: $\quad$ **for** $n \in S_i$ **do**
8: $\quad\quad Train(M_{i-1}, D_n) = M_i^{(n)}$
9: $\quad M_i = \frac{1}{b} \sum_{n \in S_i} M_i^{(n)}$
10: $\quad \omega_i = Acc(M_i, D_{N+1}) - Acc(M_{i-1}, D_{N+1})$
11: $\quad$ **if** $i > 1$ and $\omega_i > \omega_{i-1}$ **then**
12: $\quad\quad$ **for** $n \in S_i$ **do** $\varphi_n \leftarrow \varphi_n + 1$
13: $\quad\quad$ **for** $n \in S_{i-1}$ **do** $\varphi_n \leftarrow \varphi_n - 1$
14: $\quad$ **if** $\omega_i < 0$ **then**
15: $\quad\quad$ **for** $n \in S_i$ **do** $\varphi_n \leftarrow \varphi_n - 1$

We present the pseudo-code of the whole process in Algorithm 1. We split the dataset randomly into $N + 1$ parts (line 1), representing the N datasets of the participants and the test set D_{N+1}, to determine the quality of the aggregated updates. The splitting is done in a way that the resulting sub-datasets are i.i.d.; otherwise, the splitting itself would introduce some quality difference between the participants. Next, we artificially create different quality datasets using Eq. 4 (line 3). This is followed by FL (line 5-9). Round-wise improvements are captured by ω (declared in line 11 using the accuracy difference of the current and previous models). Quality scores $(\varphi_1, \ldots, \varphi_N)$ are updated in the ith round with ± 1 each time one of the three scoring rules is invoked (line 12, 13, and 15 for *The Good*, *The Bad*, and *The Ugly*, respectively).

Datasets, ML Models and Experiment Setup. For our experiments, we used the MNIST (Deng 2012) and the CIFAR10 (Krizhevsky, Nair, and Hinton 2014) datasets. MNIST corresponds to the simple task of digit recognition. It contains $70,000$ hand-written digits in the form of 28×28 gray-scale images. CIFAR10 is more involved, as it consists of $60,000$ 32×32 colour images of airplanes, automobiles, birds, cats, deers, dogs, frogs, horses, ships, and trucks. For MLP, we used a three-layered structure with hidden layer size 64, while for CNN, we used two convolutional layers with 10 and 20 kernels of size 5×5, followed by two fully-connected hidden layers of sizes 120 and 84. For the optimizer we used SGD with learning rate 0.01 and drop out rate 0.5. The combination of the two datasets and the two neural network models yield four use-cases. In the rest of the paper, we will refer to these as MM for MLP-MNIST, MC for MLP-CIFAR10, CM for CNN-MNIST, and CC for CNN-CIFAR10.

We ran all the experiments for 100 rounds and with three different FL settings, corresponding to 5, 25, and 100 participants where 2, 5, and 10 of them are selected in each round, respectively. The three FL settings combined with the four use-cases result in twelve evaluation scenarios visible in most of the Figures. We ran every experiment 10-fold, with randomly selected participants.

Empirical Quality Scores. The round-wise accumulated quality scores utilizing all three scoring rules in the twelve FL scenarios are presented in Fig. 2. The lighter shades correspond to participants with higher IDs (i.e., less added noise according to Eq. 4), while the darker shades mark low ID participants with lower quality datasets.

It is visible that the more rounds have passed, the better our scoring rules differentiate the participants; hence, it is expected that quality inference keeps improving with time. Note, that even for the participant with the highest dataset quality (i.e., the lightest curve) the quality score is rather deceasing. This is an expected characteristic of the scoring rules as there is only one rule increasing the score (*The Good*), while two decreasing it (*The Bad* and *The Ugly*). These three heuristic scoring rules combined recover the original individual dataset quality order quite well.

In Fig. 3 we show the quality scores for each individual participant. The dot marks the mean, the thick black line corresponds to the standard deviation, while the thin gray line shows the minimum and maximum values across the 10-fold experiments. In case of few participants, the trend of the quality scores is more visible: the score increases with the participant ID, i.e., participant 3 scores higher than participant 2. This is in line with the ground truth based on Eq. 4; yet the score differences are not linear as we would expect.

It is hard to evaluate the accuracy of the quality inference purely based on Fig. 2 and 3. For this reason we utilize the Spearman coefficient introduced in Eq. 3. These r_s values for the 12 studied scenarios are presented in Fig. 1. Note, that the value of the baseline (i.e., random order) is zero, and positive values indicate correlation. It is clear that the inferred quality ordering significantly deteriorates with more participants. Yet, as coefficients are always positive, the three simple rules significantly improve the baseline random guess, even with a large set of contributors. Moreover, in this paper we only considered 100 rounds of gradient updates; in many real-world applications this number would be considered low. In fact, for other realistic use-cases, quality inference is expected to perform even better, due to the availability of more information.

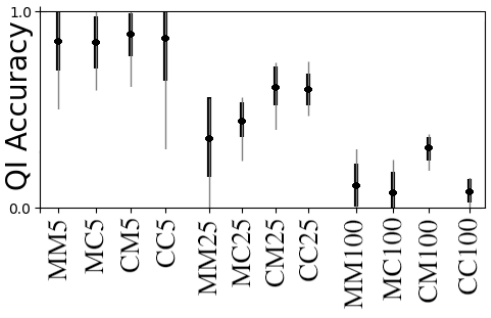

Figure 1: Spearman coefficient for the 12 scenarios.

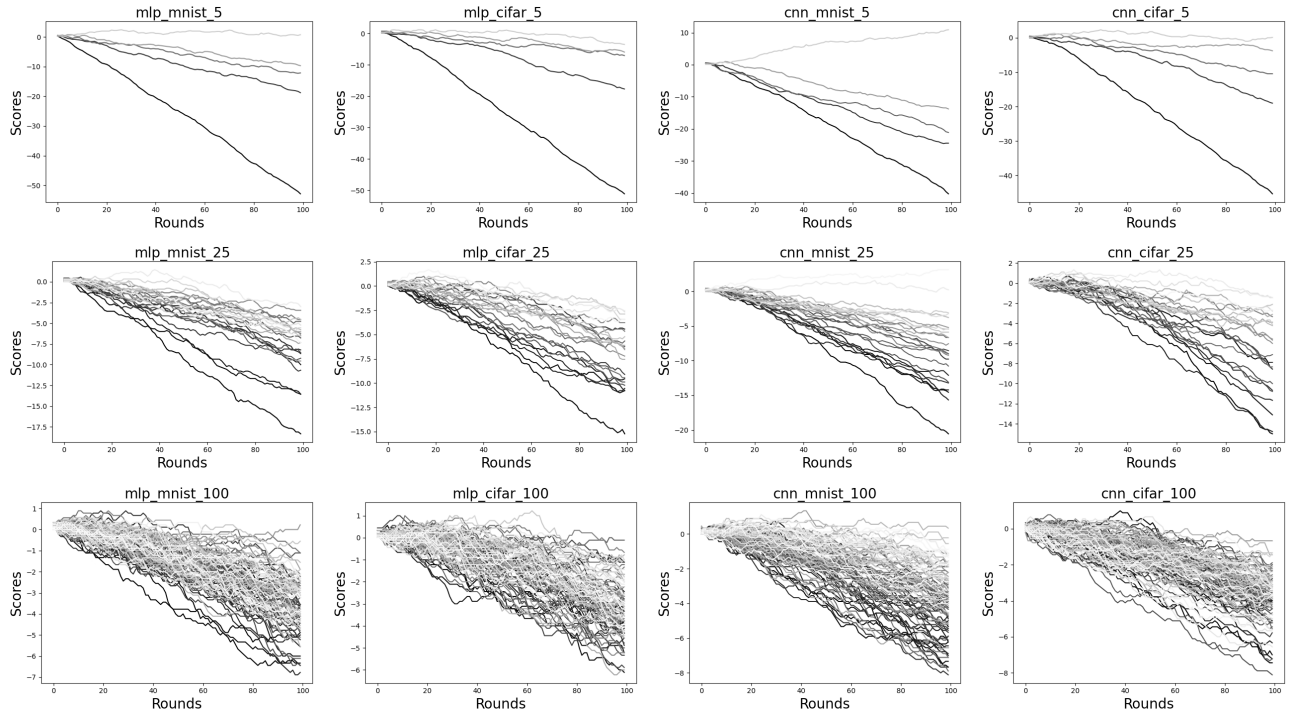

Figure 2: The average round-wise change of the participants' scores. From left to right: MM, MC, CM, and CC. From top to bottom: 5, 25, and 100 participants. The lighter the better (the darker the worse) corresponding dataset quality.

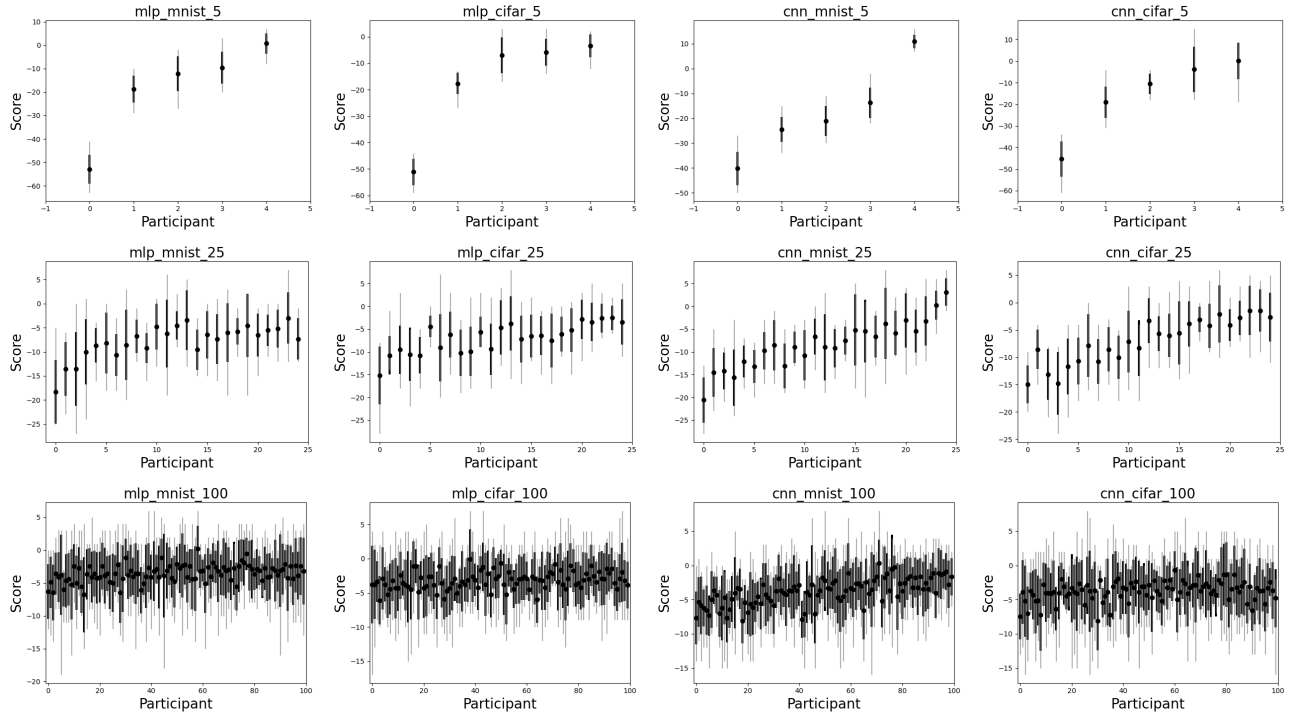

Figure 3: Quality scores of the participants. From left to right: MM, MC, CM, and CC. From top to bottom: 5, 25, and 100 participants with IDs shown on x axis where lower number correspond to lower quality dataset.

Mitigation. Note, that this quality information leakage is not by design; this is a bug, rather than a feature in FL. The simplest and most straightforward way to mitigate this risk is to use a protocol where every participant contributes in each round (incurring a sizable communication overhead). Another option is to limit access to these updates, e.g., by broadcasting the aggregated model only to the selected participants for the next round. Yet another approach is to hide the participants' IDs (e.g., via mixnets (Danezis 2003) or MPC (Goldreich 1998)), so no-one knows which participant contributed in which round except for the participants themselves. Finally, the aggregation itself could be done in a differentially private manner as well, where a carefully calculated noise is added to the updates in each round. Client-level DP (Geyer, Klein, and Nabi 2017) would by default hide the dataset quality of the participants, although at the price of requiring large volumes of noise, and therefore, low utility.[5]

Fine-tuning. We consider four techniques to improve the accuracy of quality inference, r_s, which fine-tune the parameters of our mechanism: rule combination, thresholding, using actual improvement values, and round skipping.

- Rule combination: we apply all possible combination of scoring rules in order to find which one obtains the highest accuracy.
- Thresholding: we consider using a threshold for the scoring rules, i.e., *The Ugly* only applies when the improvement is below some negative value (instead of < 0), while *The Good*/ *The Bad* applies if the improvement difference is above/below such a threshold, respectively.
- Actual improvement values: we consider alternative rule variants where the improvement values are used instead of ± 1 to capture a more fine-grained quality differentiation.
- Round skipping: In the early rounds the model does improve almost independently of the datasets, therefore we consider discarding the information from the first few rounds.

In Fig. 4 we visualize the difference between the accuracy of the original and the fine-tuned quality inference mechanism. Fine-tuning was done through a grid search with the following values: $[0, 1, \ldots, 10]$ for round skipping and $[0, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1.28, 2.56]$ for thresholding. We tried these parameters with both the actual improvement values and ± 1 counts as scores for all rule combinations.

The improvements obtained were minor, meaning that the original rules were already quite effective. Moreover, such fine-tuning would require the knowledge of the distributions of the datasets, hence the attacker cannot utilize it without relevant background knowledge. Consequently, in the applications below, we use the original rules without any fine-tuning.

Misbehaviour and Free-Rider Detection. One possible application of the inferred dataset quality information is misbehaviour detection. Here we consider both i) attackers and ii) free-riders. Their goal is either i) to decrease the accuracy of the aggregated model, or ii) to benefit from the aggregated model without contributing, respectively. We simulate these cases by computing the additive inverse of the correct gradients and using zero as the gradient. We select one such client, and use the original dataset labels without perturbation (i.e., not applying Eq. 4) for the rest of the participants. Our findings are presented in Fig. 5. In most scenarios, the position of the attacker/free-rider is in the bottom part; these preliminary results suggest that quality inference steadily outperforms the baseline random guess in misbehaviour detection.

Boosting Training Accuracy. Another potential use case for the inferred dataset quality information is accuracy boosting: based on data quality, it is expected that both training speed and the accuracy obtained could be improved when putting more emphasis on high-quality inputs. Hence, we consider weighting the updates of participants based on their quality scores. We adopt a multiplicative weight update approach (Arora, Hazan, and Kale 2012), which multiplies the weights with a fixed rate κ. Similarly to Algorithm 1, each time one of the three scoring rules is invoked the weights (initialized as $[1, \ldots, 1]$) are updated in the ith round with $\times(1 \pm \kappa)$, and then these weights are

[5]Note that studying the effects of these techniques are out of scope for this work.

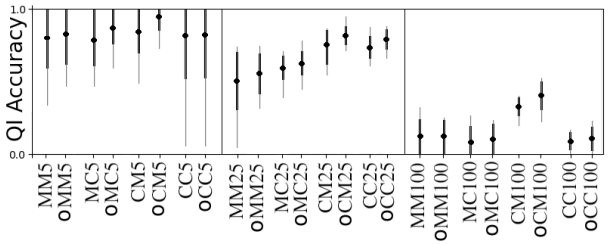

Figure 4: The original Spearman coefficient vs. the fine-tuned coefficient (marked with 'o') for the 4 use-cases with 5, 25, and 100 participants.

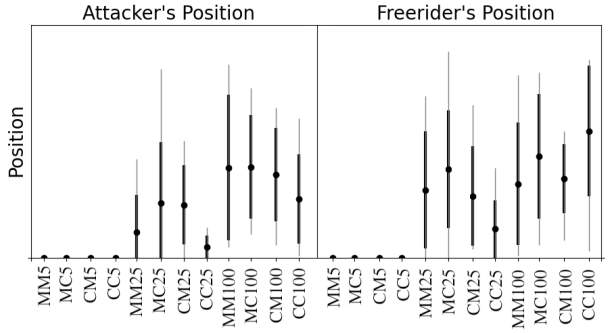

Figure 5: Position of attackers (left) and free-riders (right) for the twelve use-cases after hundred training rounds. The higher/lower results correspond to higher/lower inferred quality-wise ranks.

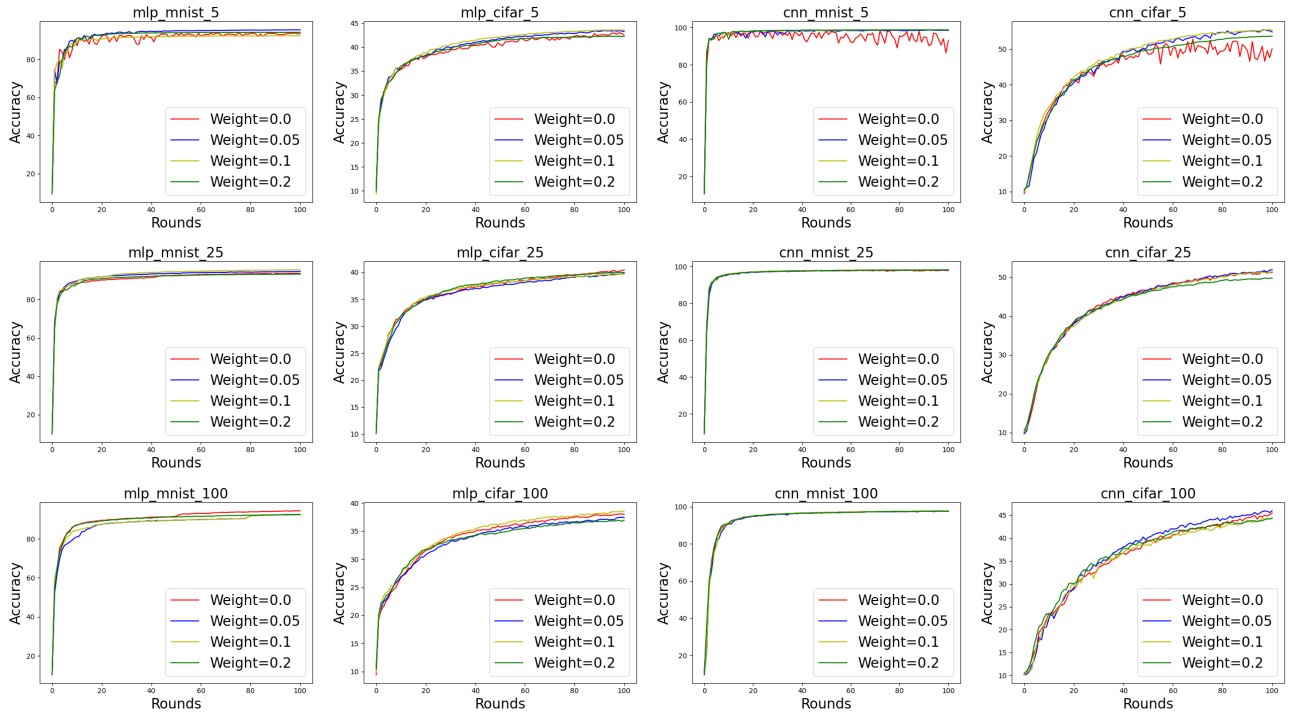

Figure 6: The round-wise accuracy of the trained models with various weights. From left to right: 5, 25, and 100 participants. From top to bottom: MM, MC, CM, and CC.

used during aggregation. For our experiments, we set $\kappa = \{0.00, 0.05, 0.10, 0.20\}$, where the first value corresponds to the baseline without participant weighting. We present our results in Fig. 6. It is conclusive that using weights based on our scoring rules i) mitigates the effect of low quality datasets as the training curves are smoother and ii) improves the original accuracy.

Related Work

In this section we briefly present related research efforts, including but not limited to simple scoring mechanisms, well known privacy attacks against machine learning, and data quality. The theoretical analysis of quality inference does relate to (Dinur and Nissim 2003) as attempting to reconstruct the dataset quality order is similar to reconstructing the entire dataset based on query outputs.

Participant Scoring. Simple but effective scoring rules are prevalent in the field of complex ICT-based systems, especially characterizing quality. For instance binary or counting signals can be utilized i) to steer peer-to-peer systems measuring the trustworthiness of peers (Kamvar, Schlosser, and Garcia-Molina 2003), ii) to assess and promote content in social media (Van Mieghem 2011), iii) to ensure the proper natural selection of products on online marketplaces (Lim et al. 2010), and iv) to select trustworthy clients via simple credit scoring mechanisms (Thomas, Crook, and Edelman 2017).

A free-rider detection mechanism for collaborative learning are presented in (Lin, Du, and Liu 2019; Fraboni, Vidal, and Lorenzi 2021). In contrast, (Liu et al. 2021) proposes an online evaluation method that also defines each participant's impact based on the current and the previous rounds. Although their goal is similar to ours, we consider *SA* being utilized, while neither of the above mechanisms are applicable in such a case. (So et al. 2021) considers the reconstruction of the participation matrix by simulating the same round several times with different participants. Instead, we assume such participation information is available, and we emulate the training rounds by properly updating the model. Similarly to out first application (Chen et al. 2020) weights the participants based on their data quality using the cross-entropy of the local model predictions. Their experiments consider only five participants and two quality classes (fully correct or incorrect), while we study more informative, fine-grained quality levels with both smaller and larger sets of participants.

Privacy Attacks. There are several indirect threats against FL models. These could be categorized into model inference (Fredrikson, Jha, and Ristenpart 2015), membership inference (Shokri et al. 2017), parameter inference (Tramèr et al. 2016), and property inference (Ganju et al. 2018; Melis et al. 2019). Our quality inference could be considered as an instance of the last. Another property inference instance is the quantity composition attack (Wang et al. 2019a), which aims to infer the proportion of training labels among the participants in FL. This attack is successful even with *SA* protocols or under the protection of differential privacy; in contrast to our work they focus on inferring the different distribu-

tions of the datasets, while we aim to recover the relative quality information on i.i.d. datasets. Finally, (Wang et al. 2019b) also attempts to explore user-level privacy leakage within FL. Similarly to our work, the attack defines client-dependent properties, which then can be used to distinguish the clients from one another. They assume a malicious server utilizing a computationally heavy GAN for the attack, which is the exact opposite of our honest-but-curious setup with limited computational power.

Privacy Defenses. Quality inference can be considered as a property inference attack, hence, naturally it can be "mitigated" via client-level differential privacy (Geyer, Klein, and Nabi 2017). Moreover, as we simulate different dataset qualities with the amount of added noise, what we want to prevent is the leakage of the added noise volume. Consequently, this problem also relates to private privacy parameter selection, as label perturbation (Papernot et al. 2016) (which we use to mimic different dataset quality levels) is one technique for achieving differential privacy (Desfontaines and Pejó 2020). Although some works set the privacy parameter using economic incentives (Hsu et al. 2014; Pejo, Tang, and Biczok 2019), we are not aware of any research which considers defining the privacy parameter itself also privately.

Data Quality. In this work we naively assume that data quality is in a direct relation with noise present in the data. Naturally, this is a simplification: there is an entire computer science discipline devoted to data quality; for a comprehensive view on the subject we refer the reader to (Batini, Scannapieco et al. 2016).

A complementary notion is the Shapley value (Shapley 1953) which was designed to allocate goods to players proportionally to their contributions (which can be interpreted as input data quality in FL). The main drawback of this payment distribution scheme is that it is computationally not feasible in most scenarios. Several approximation methods were proposed in the literature using sampling (Castro, Gómez, and Tejada 2009), gradients (Ghorbani and Zou 2019; Kwon and Zou 2021; Nagalapatti and Narayanam 2021) or influence functions (Koh and Liang 2017; Xue et al. 2020; Xu, van der Maaten, and Hannun 2020). Although some are promising, all previous methods assume explicit access to the datasets or the corresponding gradients. *Consequently, these methods are not applicable when SA is enabled during FL.* Our quality inference rules can be considered as a first step towards a contribution score when *no information on individual datasets is available*.

Conclusion

Federated learning is the most popular collaborative learning framework, wherein each round only a subset of participants updates a joint machine learning model. Fortified with secure aggregation only aggregated information is learned both by the participants and the server. Yet, in this paper we devised few simple quality scoring rules that were able to successfully recover the relative ordering of the participant's dataset qualities; even when secure aggregation is in use. Our method neither requires any computational power (such as shadow models), nor any background information besides a small representative dataset (or access to an evaluation oracle) in order to be able to evaluate the improvement of model accuracy after each round.

Through a series of image recognition experiments we showed that it is possible to restore the relative quality-wise ordering with reasonably high accuracy. Our experiments also revealed a connection between the accuracy of the quality inference and the complexity of the task and the used architecture. What is more, we done an ablation study which suggest the basic rules are almost ideal. Lastly, we demonstrated how quality inference could i) boost training efficiency by weighting the participants and ii) detect misbehaving participants based on their quality score.

Limitations and Future Work. This paper has barely scratched the surface of quality inference in federated learning using only the aggregated updates. We foresee multiple avenues towards improving and extending this work, e.g., using machine learning techniques to replace our naive rules or by relaxing attacker constraints concerning computational power and background knowledge. For the sake of clarity, we have restricted our experiments to visual recognition tasks, so it is an open question whether our rules does generalize to other domains as well.

Another interesting direction is to use quality inference for computing contribution scores for the participants, and approximate their Shapley value (Shapley 1953). We are not aware of any scheme capable of doing this *when secure aggregation is enabled* besides the conceptual idea in (Pejó, Biczók, and Ács 2021). Finally, the personal data protection implications of the information leakage caused by quality inference is also of interest: could such quality information be considered private (and consequently should be protected)? This issue might have practical relevance to federated learning platforms already available.

References

Arora, S.; Hazan, E.; and Kale, S. 2012. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2938–2948. PMLR.

Batini, C.; Scannapieco, M.; et al. 2016. Data and information quality. *Cham, Switzerland: Springer International Publishing. Google Scholar*.

Castro, J.; Gómez, D.; and Tejada, J. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & Operations Research*, 36(5): 1726–1730.

Chen, Y.; Yang, X.; Qin, X.; Yu, H.; Chen, B.; and Shen, Z. 2020. Focus: Dealing with label quality disparity in federated learning. *arXiv preprint arXiv:2001.11359*.

Danezis, G. 2003. Mix-networks with restricted routes. In *International Workshop on Privacy Enhancing Technologies*. Springer.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*.

Desfontaines, D.; and Pejó, B. 2020. Sok: Differential privacies. *Proceedings on Privacy Enhancing Technologies*, 2020(2): 288–313.

Diaconis, P.; and Graham, R. L. 1977. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society: Series B (Methodological)*.

Dinur, I.; and Nissim, K. 2003. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM.

Fraboni, Y.; Vidal, R.; and Lorenzi, M. 2021. Free-rider attacks on model aggregation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, 1846–1854. PMLR.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*.

Ganju, K.; Wang, Q.; Yang, W.; Gunter, C. A.; and Borisov, N. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 619–633.

Gentry, C.; et al. 2009. *A fully homomorphic encryption scheme*, volume 20. Stanford university Stanford.

Geyer, R. C.; Klein, T.; and Nabi, M. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Ghorbani, A.; and Zou, J. 2019. Data shapley: Equitable valuation of data for machine learning. *arXiv preprint arXiv:1904.02868*.

Goldreich, O. 1998. Secure multi-party computation. *Manuscript. Preliminary version*.

Harville, D. 1976. Extension of the Gauss-Markov theorem to include the estimation of random effects. *The Annals of Statistics*.

Hsu, J.; Gaboardi, M.; Haeberlen, A.; Khanna, S.; Narayan, A.; Pierce, B. C.; and Roth, A. 2014. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE.

Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; and Zhang, X. 2021. Source Inference Attacks in Federated Learning. *arXiv preprint arXiv:2109.05659*.

IMDB. 1966. The Good, the Bad and the Ugly. https://www.imdb.com/title/tt0060196/.

Kamvar, S. D.; Schlosser, M. T.; and Garcia-Molina, H. 2003. Incentives for combatting freeriding on P2P networks. In *European Conference on Parallel Processing*, 1273–1279. Springer.

Kerkouche, R.; Ács, G.; and Castelluccia, C. 2020. Federated Learning in Adversarial Settings. *arXiv preprint arXiv:2010.07808*.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730*.

Konečný, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016. Federated Learning: Strategies for Improving Communication Efficiency. *arXiv:1610.05492 [cs]*.

Krizhevsky, A.; Nair, V.; and Hinton, G. 2014. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*.

Kwon, Y.; and Zou, J. 2021. Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning. arXiv:2110.14049.

Lim, E.-P.; Nguyen, V.-A.; Jindal, N.; Liu, B.; and Lauw, H. W. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 939–948.

Lin, J.; Du, M.; and Liu, J. 2019. Free-riders in Federated Learning: Attacks and Defenses. *arXiv preprint arXiv:1911.12560*.

Liu, B.; Yan, B.; Zhou, Y.; Wang, J.; Liu, L.; Zhang, Y.; and Nie, X. 2021. FedCM: A Real-time Contribution Measurement Method for Participants in Federated Learning. *arXiv preprint arXiv:2009.03510*.

Ludeman, L. C. 2003. *Random processes: filtering, estimation, and detection*. John Wiley & Sons, Inc.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

Melis, L.; Song, C.; De Cristofaro, E.; and Shmatikov, V. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE.

Nagalapatti, L.; and Narayanam, R. 2021. Game of Gradients: Mitigating Irrelevant Clients in Federated Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9046–9054.

Papernot, N.; Abadi, M.; Erlingsson, U.; Goodfellow, I.; and Talwar, K. 2016. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*.

Pejó, B.; Biczók, G.; and Ács, G. 2021. Measuring Contributions in Privacy-Preserving Federated Learning. *ERCIM NEWS*, 35.

Pejo, B.; Tang, Q.; and Biczok, G. 2019. Together or Alone: The Price of Privacy in Collaborative Learning. *Proceedings on Privacy Enhancing Technologies*.

Shapley, L. S. 1953. A value for n-person games. *Contributions to the Theory of Games*.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE.

So, J.; Ali, R. E.; Guler, B.; Jiao, J.; and Avestimehr, S. 2021. Securing Secure Aggregation: Mitigating Multi-Round Privacy Leakage in Federated Learning. *arXiv preprint arXiv:2106.03328*.

Thomas, L.; Crook, J.; and Edelman, D. 2017. *Credit scoring and its applications*. SIAM.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium (USENIX Security 16)*.

Van Mieghem, P. 2011. Human psychology of common appraisal: The Reddit score. *IEEE Transactions on Multimedia*, 13(6): 1404–1406.

Wang, L.; Xu, S.; Wang, X.; and Zhu, Q. 2019a. Eavesdrop the Composition Proportion of Training Labels in Federated Learning. *arXiv:1910.06044 [cs, stat]*.

Wang, Z.; Song, M.; Zhang, Z.; Song, Y.; Wang, Q.; and Qi, H. 2019b. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 2512–2520. IEEE.

Xu, M.; van der Maaten, L.; and Hannun, A. 2020. Data Appraisal Without Data Sharing. arXiv:2012.06430.

Xue, Y.; Niu, C.; Zheng, Z.; Tang, S.; Lv, C.; Wu, F.; and Chen, G. 2020. Toward Understanding the Influence of Individual Clients in Federated Learning. *arXiv preprint arXiv:2012.10936*.

Zar, J. H. 2005. Spearman rank correlation. *Encyclopedia of Biostatistics*, 7.