On the Federated Detection of Hate Speech

Valentin Gorbunov[1], **Emiliano De Cristofaro**[1]
[1]University College London
{v.gorbunov, e.decristofaro}@ucl.ac.uk

Abstract

The increasing amount of hate speech on social media and specifically the challenging task of automatic hate speech detection has drawn the interest of researchers. Existing studies primarily focus on classifiers to automatically detect hate speech, and annotated corpora that can be used to study the problem. However, with the rising concern around user privacy, data is increasingly fragmented and difficult to share. Thus, organisations are limited in their ability to fully exploit user data. Hence, we design and evaluate a system that allows participants to exploit such siloed data to build effective hate speech detection models. We do this by applying a technique called Federated Learning. We find that the federated setting poses significant challenges, in terms of performance, and the more Independently and Identically Distributed (IID) the data distribution among participants, the more an organisation benefits from participating in our system.

Introduction

While the rise of user-generated content on Web 2.0 platforms has availed the democratization of content production, it has also harbored potential cyber safety issues. Of increasing concern is the issue of hate speech; the task of automatic hate speech detection has gained traction in the AI community, as recent surveys attest (Poletto et al. 2021; Fortuna and Nunes 2018; Schmidt and Wiegand 2019; Lucas 2014). Existing studies on automatic hate speech detection (Davidson et al. 2017; Aluru et al. 2020) employ centralized machine learning pipelines, which require that all training data be stored at a centralized location, where the workflow it takes to produce a model is performed.

However, the increasingly distributed nature of user data compounded with the issue of class imbalance in hate speech datasets (Aroyehun and Gelbukh 2018) means that organisations don't generate enough data of their own to derive unbiased insights. Therefore, there is a growing need to share data. Nevertheless, as data protection regulations become more stringent and fragmented, the respective obligations and liabilities of those involved in processing sensitive hate speech data are growing; thus training data is increasingly distributed across data silos. To develop effective hate speech detection models, there is a need to exploit such distributed databases without exchanging the raw data. To this end, we investigate Federated Learning, an approach that allows participants to learn a shared model by aggregating locally computed updates, leaving the training data distributed and thereby preserving its privacy (McMahan et al. 2017).

The objective of this paper is to design a federated learning system to detect hate speech and evaluate it in terms of its utility, to determine whether organizations would benefit from participating. Accordingly, we design experiments to evaluate its performance, in the context of the unbalanced and non-IID data distributions that are defining characteristics of the federated setting.

Background

Data Protection

Data protection has become a prime focus in policy (Koops 2014). With that, there have been a number of studies assessing the impact of data protection legislation on AI. The European Parliamentary Research Service's (EPRS) study on the impact of the General Data Protection Regulation (GDPR) on AI discussed the controller obligations concerning AI systems involved with automated decision-making, such as automated hate speech detection (Sartor and Lagioia 2020). It indicated that providing guidance on the GDPR requires a multilevel approach involving all stakeholders, from data protection authorities to civil society; and conceded that there is a broad, ongoing debate to determine the scope of data subject's rights that apply to AI processing of personal data. These complexities may 'needlessly hamper the development of AI applications' (Sartor and Lagioia 2020). Furthermore, in the case of the judgement of 24 September 2019, Google Inc. v. Commission Nationale de l'Informatique et des Libertés (CNIL), the Court held that there is potential for fragmentation in the level of protection amongst Member States; affirming that it's for Member States to make derogations (Samonte 2020). Thus, it repeated previous evaluations, that found the GDPR to be indeterminate at the practical level (Koops 2014). A similar conclusion was drawn in a review of The Data Protection Directive (DPD) (Robinson et al. 2009).

The effectiveness of automatic hate speech detection systems relies on the availability of a large volume of train-

ing data. Nevertheless, fragmented and indeterminate data protection regulations impede data flows across organizational/geographical borders. For instance, Facebook's EU-U.S. data flows have come under threat after Ireland's High Court dismissed Facebook's challenge over a regulatory inquiry, in which the Court of Justice of the European Union (CJEU) annulled an EU-U.S. data transfer agreement for the second time in the past five years (Murphy 2022).

To help meet their data protection obligations, organisations commonly use data anonymization and de-identification techniques; the UK Information Commissioner's Office publishes a code of practice on anonymization, aimed at data controllers (Graham 2012). Proper data anonymization exempts controllers from the ambit of the GDPR (Sartor and Lagioia 2020). However, similarly to the DPD, the GDPR is based on a contextual definition of personal data. That is to say that personalization is seen as a property of the data's environment, as opposed to characterized on the face of the data itself (Stalla-Bourdillon and Knight 2016). The implication is that the anonymization approach has to be dynamic. In this regard, Arcticle 29 of the Data Protection Working Party puts forward that in order for data to be considered personal, there must be a 'content', 'purpose' or 'result' element present (Party 2007). A content element is present if it's possible to identify an individual from the data itself. Additionally, when processing the data may influence the status or behaviour of an individual, the data is considered personal data by purpose, even without a content element. Furthermore, if an individual may be treated differently as a result of data processing, the data is considered personal data by result, regardless of the presence of other elements. On account of being an automated decision-making system, it's unlikely that an automated hate speech detection system could accommodate such a holistic approach. Furthermore, traditional anonymization techniques may cause data to lose its utility (Bioglio and Pensa 2022). This is particularly the case with hate speech data, which is targeted towards a specific person/group and context dependant (Poletto et al. 2021).

Automatic Hate Speech Detection

Several recent workshops have helped to establish automatic hate speech detection as a challenging task (Pelicon, Martinc, and Novak 2019). The most competitive model in the shared task on aggression identification, from the first workshop on Trolling, Aggression and Cyberbullying (TRAC-1), achieved a weighted F-score of just 0.64, for both Hindi and English Facebook test sets (Kumar et al. 2018).

Hate speech detection has primarily been approached as a supervised learning problem (Schmidt and Wiegand 2019), relying on manually engineering features that are subsequently consumed by classic classification algorithms. Of the classification algorithms, Support Vector Machines (SVM), Recurrent neural networks and Logistic Regression have proved to be particularly popular (Pelicon, Martinc, and Novak 2019). This approach has been facilitated by the vast amount of user-generated content available on social media, specifically on microblogging platforms (Poletto et al. 2021). The most common type of resource from which features are constructed is annotated corpora, meant as a collection of labelled textual instances from a variety of sources. They are often developed specifically for training an automated detection system and presented jointly with an evaluation of said system (Poletto et al. 2021). The mircroblogging platform Twitter is the most exploited source, due to the short maximum allowable length of a tweet and its data sharing policy (Poletto et al. 2021).

Nonetheless, existing studies have found that some annotated tweets are no longer available from Twitter due to their offensive content (Pereira-Kohatsu et al. 2019). This raises an important issue concerning the scope of the right to erasure with regards to AI-based processing. Erasing data used in constructing a hate speech detection model makes it difficult, if not impossible, to reproduce results and determine the validity of the model (Sartor and Lagioia 2020). Furthermore, it reduces the volume of data available to organisations for training their own hate speech detection systems.

Additionally, imbalanced class distributions, that pervade existing hate speech corpora (Aroyehun and Gelbukh 2018; Pelicon, Martinc, and Novak 2019), further exacerbate the challenge for organisations. (Pelicon, Martinc, and Novak 2019) noticed that the class imbalance in their hate speech datasets had a significant impact on the performance of their systems. To curb the impact of the class imbalance, they resampled the dataset by randomly removing instances from the majority classes, leading to an improvement in performance. Be that as it may, the effect of the class imbalance persevered.

Federated Learning

Taxonomy Federated learning typically refers to model-centric federated learning. The aim of model-centric federated learning is to train a globally shared machine learning model using multiple local datasets, without exchanging local data samples (Śmietanka, Pithadia, and Treleaven 2020). Rather, the global model is updated by aggregating the parameters of local models, such as the coefficients and intercepts of a logistic regression model. In contrast, data-centric federated learning is a nascent application that allows third parties to make requests for training or inference against an organisation's data, while preserving the data's privacy (Śmietanka, Pithadia, and Treleaven 2020).

The design and deployment of distributed applications, such as federated learning, are supported by a framework of tools and services within a distributed environment (Bauer et al. 1994). With regards to distributed application architecture, there are two types of federated learning. In centralized federated learning, a central server orchestrates the federated learning algorithm; the server is responsible for selecting clients for participation and aggregating model updates (Kairouz et al. 2021). In contrast, clients participating in decentralized federated learning coordinate themselves; updates are exchanged peer-to-peer (Kairouz et al. 2021). Although a decentralized architecture helps to alleviate bottlenecks, such as that which may occur at the central server, additional consideration must be had for the network topology (Kairouz et al. 2021). The tools used in the dis-

tributed environment further differentiate the types of federated learning. Cross-silo federated learning exploits data residing in siloed data centers , that are distributed across organisational/geographical borders (Kairouz et al. 2021). On the other-hand, in cross-device federated learning, learning takes place on edge devices, such as mobile phones. For cross-device federated learning, clients are massively distributed (Kairouz et al. 2021). Furthermore, only a fraction of clients may be available at any given time (Reddi et al. 2020). Additionally, for cross-device supervised learning tasks, labels should be able to be inferred naturally from user interaction (McMahan et al. 2017). Conversely, for cross-silo federated learning, the massively distributed property does not hold and clients have a high availability, with all clients being almost always available (Kairouz et al. 2021).

Moreover, the nature in which data is partitioned across clients affects the training architecture (Kairouz et al. 2021). In this regard, we can further differentiate between types of federated learning. In horizontal federated learning, sometimes referred to as homogeneous federated learning, clients share the same feature space but have different sample spaces (Yang et al. 2019). That is to say that the data is horizontal. In comparison, in vertical federated learning, sometimes referred to as heterogeneous federated learning, clients have different sets of features; thus a different training architecture is used, in which clients exchange specific intermediate results rather than model parameters (Yang et al. 2019).

Algorithm The optimization problem implicit in federated learning is known as the federated optimization problem (McMahan et al. 2017). Federated learning is the iterative process of solving this problem, over multiple federated learning rounds. Many recent studies have relied on a specific optimization method, primarily Gradient Descent or Stochastic Gradient Descent, to be used on all clients (Li et al. 2020). However, any iterative optimization method (Kelley 1995), that converges to a solution on some specified class of problems, is applicable to the federated optimization problem. (Li et al. 2020) even present a method agnostic framework, wherein different clients can use various optimization methods.

To illustrate the algorithm, consider how Gradient Descent may be applied to the federated optimization problem. Given a fixed set of K clients, over which a horizontal dataset of size n is partitioned; with $\mathcal{P}_k$ being the fixed set of samples on client k, and $n_k = |\mathcal{P}_k|$; the central server selects a C-fraction of clients to participate in each round, where C is the global batch size. Taking $C = 1$, which corresponds to batch optimization; client k makes E passes over its local dataset for each federated learning round, where E is the number of local epochs; and updates its local model multiple times

$$w^k \leftarrow w^k - \eta \nabla F_k(w^k) \tag{1}$$

where w^k is the local model. The central server then aggregates the updated models by taking their weighted average

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k \tag{2}$$

where w_{t+1} is the global model at round $t+1$. This approach is known as $FederatedAveraging$, or $FedAvg$.

In contrast to distributed learning, which aims to increase the scale and speed of model training through parallelizing computation, as in (Dean et al. 2012), federated learning aims to train on heterogeneous datasets. In this vein, the key properties that differentiate the federated setting are:

- Unbalanced data, i.e., local datasets vary in size, potentially spanning orders of magnitude.
- The partitions are not formed by distributing the dataset across clients uniformly at random, i.e., the IID assumption

$$\mathbb{E}_{\mathcal{P}_k}[L_k(w)] = \sum_{k=1}^{K} \frac{n_k}{n} L_k(w) \tag{3}$$

  where $\mathbb{E}_{\mathcal{P}_k}[L_k(w)] = \mathbb{E}[L_k(w)|\mathcal{P}_k]$ and $L_k(w)$ is the average loss of the global model w on samples in $\mathcal{P}_k$, does not hold (McMahan et al. 2017).

Methodology

We now explain our methodology for selecting a hate speech classifier, designing a federated learning system, and evaluating the system in terms of it's utility.

Selecting a Hate Speech Classifier

Since it would be impossible to find and federate every hate speech classifier published, we opt to select one of them. Specifically, we select a classifier that uses a modeling method and features that we believe to be representative of the wider literature. We conducted a survey of the latest hate speech detection models. We built on the surveys by (Anon 2020) and (Poletto et al. 2021). Additionally, we performed an independent literature search. We collected academic works found on Google Scholar and Paperswithcode. We limited the search results from Google Scholar to the first ten pages for each keyword [1] and sorted results by relevance, with a time filter of 'since 2017'. In our criteria, we adopted the same definition of hate speech as (Poletto et al. 2021). We briefly summarize it here, as content inciting violence towards, or threatening the safety or dignity of, a protected group, or an individual belonging to such a group, due to their membership in said group and not due to personal attributes. Hate speech overlaps with related concepts, such as misogyny and racism, but they are not exactly the same (Poletto et al. 2021). We focused on studies that used English datasets. We adopted 4 criteria to trim down our results: 1) The classifier must be from a study published in or after 2017, so it's representative of the latest literature, 2) The code is open source and publicly accessible, 3) We can validate the results, 4) There is extensive literature on how to federate the classifier.

[1] Keywords: hate speech detection, automated hate speech detection, hate speech detection using machine learning, hate speech detection using deep learning, survey of automated hate speech detection, hate speech detection using neural networks

Our search found many studies on profiling hateful users, as opposed to detecting hate speech (Ribeiro et al. 2018; Ribeiro et al. 2017; Ahmed, Vidgen, and Hale 2022). This shows a potential ongoing shift in focus from content towards users, due to the noisiness and subjectivity of user-generated content (Ribeiro et al. 2018). Nevertheless, the scope of the studies was different to ours. Additionally, the majority of studies on deep learning classifiers focused on specific low resource languages, such as Bengali (Karim et al. 2021). These classifiers were deemed not to be representative of the wider literature. Similarly to (Fortuna and Nunes 2018), we found there to be a paucity of studies that make their code publicly accessible.

We shortlisted 2 studies from our search, that best fit our criteria, in addition to the 3 studies experimented with by (Anon 2020). Specifically, (Badjatiya et al. 2017) were the first to investigate the application of Deep Neural Networks (DNN's) to hatespeech detection. Their study examined Convolutional Neural Networks (CNN's), FastText and Long Short-Term Memory Networks (LSTM's). The DNN's outperformed state of the art methods at the time, on an annotated corpus of 16K english tweets. However, the study focused on the task of classifying a tweet as racist, sexist or neither, which made it difficult to ascertain the extent to which they were really detecting hate speech. Furthermore, it wasn't possible to validate the results, as the majority of tweets are no longer available from Twitter. (Aluru et al. 2020) investigated the performance of 3 classifiers: CNN-GRU, Logistic Regression and BERT, on a parallel corpora of diverse languages. BERT is a pre-trained, bidirectional, unsupervised language representation that has recently become popular, particularly in multilingual scenarios (Aluru et al. 2020). We focused on results on the English dataset, in the monolingual scenario. CNN-GRU was beaten by all other methods in all tests. Logistic Regression and BERT were consistently competitive. However, Logistic Regression performed better in low resource settings, whereas BERT performed better when the training size was the full training data.

We note that some of the classifiers use similar modelling methods. The classifier in (Pelicon, Martinc, and Novak 2019) and the BERT classifier in (Aluru et al. 2020) are both BERT based. Whereas, Davidson's classifier (Davidson et al. 2017), Ex Machina (Wulczyn, Thain, and Dixon 2017) and the Logistic Regression classifier in (Aluru et al. 2020) are all based on Logistic Regression. There is a plethora of literature on how to federate logistic regression. On the other-hand, there is not much in the way of approaches to federating BERT based classifiers (Liu and Miller 2020). Therefore, we opted to select Davidson's classifier, as it is highly cited and makes a distinction between hate speech and offensive language, which some other approaches have unfortunately conflated (Burnap and Williams 2016; Waseem and Hovy 2016).

Dataset

We use the dataset made available by (Davidson et al. 2017). The dataset consists of 25k tweets. Some of the tweets contain words and/or phrases from a hate speech lexicon compiled by Hatebase.org. Each tweet has been manually labelled by crowdworkers, with one of three labels: hate speech (0), offensive (1), or neither hate speech nor offensive (2). Figure 1 illustrates the imbalanced nature of the classification problem. Most tweets containing hate speech, as defined by Hatebase, were labelled as offensive by crowdworkers. Only 5% of tweets are labelled as hate speech, with more tweets considered neither hate speech nor offensive than are considered hate speech. This makes the dataset as imbalanced as comparable datasets. Approximately 5% of annotated samples from the corpus collated by (Burnap and Williams 2016) contain 'cyber hate

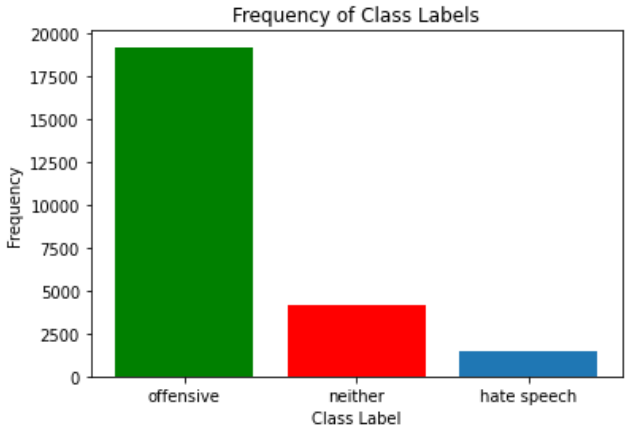


Figure 1: Imbalanced nature of the classification problem

Selected Classifier

(Davidson et al. 2017) use Logistic Regression with L2 regularization as a classifier, within their machine learning pipeline. Any iterative optimization method can be applied to the federated optimization problem. The caveat is that the solver, i.e., the routine that orchestrates model optimization and incorporates the given optimization method, must support warm-start training. As the federated optimization problem is solved over the course of multiple federated learning rounds, it can be seen as a sequence of related optimization problems. Initializing a new model for a given problem with the solution to the previous problem is known as 'warm-starting' the training of the new model, as opposed to training it from scratch (Ash and Adams 2020). For our system to converge on a solution to our federated optimization problem, it is necessary that the solver used by clients supports warm-start training. To this end, our system does away with the Liblinear solver (Fan et al. 2008) employed by (Davidson et al. 2017), which incorporates a Newton method with a one-versus-rest framework for multi-class Logistic Regression. Rather, we use scikit-learn's (Pedregosa et al. 2011) lbfgs solver, that incorporates a widely used Quasi-Newton method, L-BFGS (Liu and Nocedal 1989), together with the softmax likelihood function and its corresponding log loss function (Brownlee 2019).

Design of Federated Learning System

We design a single-machine simulation of a Federated Learning system to detect Hate Speech. Figure 2 illustrates the system's architecture. We run an entire Federated Learning system, with one central server and multiple clients on a single machine, using the Flower framework (Beutel et al. 2020) and multiprocessing. Real-world Federated Learning systems must deal with practical issues that are not present in our simulation, such as dynamically changing local datasets, security considerations and connectivity issues (McMahan et al. 2017). Nevertheless, our simulation is a good starting point for evaluating the utility of Federated Learning for hate speech detection.

Our simulation is run in a controlled environment that is suitable for experiments. We split the dataset using a stratified train/test split of 9/1. It's desirable for the train/test split to preserve the proportions of examples in each class, as observed in the full dataset, particularly when there is a class imbalance (Brownlee 2016). There is a fixed set of 10 clients, each with a fixed local dataset. The local train/test splits are formed by partitioning our train/test split horizontally across the clients. For cross-device, supervised Federated Learning tasks, labels should be able to be inferred naturally from user interaction (McMahan et al. 2017), such as in predictive text. In hate speech detection, labels can't be inferred naturally from user interaction. Therefore, supervised approaches to hate speech detection can be federated cross-silo but not cross-device. The Massively Distributed property and the Limited Communication property don't apply to cross-silo Federated Learning (Kairouz et al. 2021), thus they are outside of our scope. In this vein, all clients participate in each round of Federated Learning.

We assume a synchronous update scheme that proceeds in rounds of communication. Each Federated Learning round consists of a training round followed by an evaluation round. There are two main approaches to evaluate models in a Flower federated learning system: centralized evaluation and federated evaluation. In centralized evaluation, the global model is validated on the server-side. Whereas, in federated evaluation, the global model is validated on the client-side (Beutel et al. 2020). We use a federated evaluation approach, where the test set on each client has the same probability distribution as the client's training set, to model a real-world deployment of our federated learning system. Before Federated Learning begins, the global model is initialized with the parameters of a randomly selected client's local model; in our simulation, all local models are initialized to zero. At the beginning of each training round, the server sends the current global model parameters to each client. Subsequently, each client updates its local model with the parameters of the global model. Then, the client trains its local model on its local training data and sends the updated local model parameters to the server. Thereafter, the server aggregates these client updates, through $FedAvg$, and applies them to the global model. At the beginning of each evaluation round, the server sends the current global model parameters to each client. Thereupon, each client evaluates the model's validity on its test split. Federated Learning continues for 50 federated learning rounds, so that we can observe the system's performance. Consequently, we simulate a centralized, model-centric, cross-silo, horizontal Federated Learning system to detect Hate Speech.

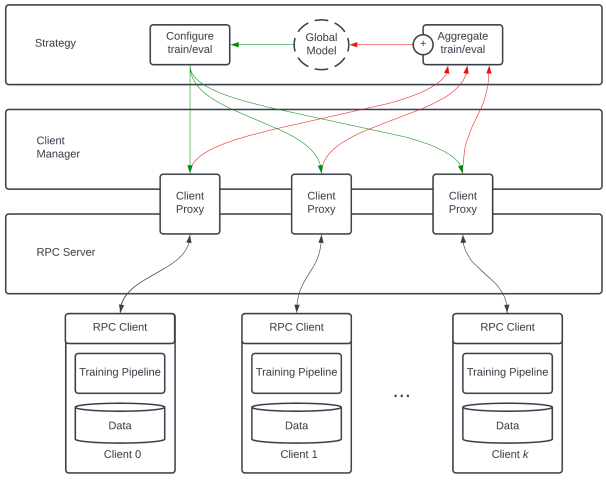


Figure 2: Macro-architecture of the system

Experimental Design

In this section, we explain how we assess our proposed Federated Learning system, in terms of its utility. First, we run our simulation under the IID data distribution case as the baseline. We then run the system on the unbalanced and non-IID data distributions characteristic of the federated setting, which existing studies often refer to as the Non-IID setting, as opposed to the IID setting. (Kairouz et al. 2019) gives a taxonomy of unbalanced and non-IID distribution cases. Consider the local discrete probability distribution, defined by the joint probability mass function $p^k_{XY}(x_i, y_i)$; where (x_i, y_i) is a sample from our full dataset, with feature vector x_i and label y_i. We rewrite $p^k_{XY}(x_i, y_i)$ in terms of the conditional distributions $p^k_{XY}(x_i|y_i)p^k_Y(y_i)$ and $p^k_{XY}(y_i|x_i)p^k_X(x_i)$, to allow us to formalise the 5 cases:

1. Label Distribution Skew
   $(\forall k_1, k_2 \in K . k_1 \neq k_2 \rightarrow p^{k_1}_Y(y_i) \neq p^{k_2}_Y(y_i))$
2. Feature Distribution Skew
   $(\forall k_1, k_2 \in K . k_1 \neq k_2 \rightarrow p^{k_1}_X(x_i) \neq p^{k_2}_X(x_i))$
3. Same labels but different features
   $(\forall k_1, k_2 \in K . k_1 \neq k_2 \rightarrow p^{k_1}_{XY}(x_i|y_i) \neq p^{k_2}_{XY}(x_i|y_i))$
4. Same features but different labels
   $(\forall k_1, k_2 \in K . k_1 \neq k_2 \rightarrow p^{k_1}_{XY}(y_i|x_i) \neq p^{k_2}_{XY}(y_i|x_i))$
5. Quantity Skew/Unbalanced data
   $(\forall k_1, k_2 \in K . k_1 \neq k_2 \rightarrow (p^k_{XY}(x_i, y_i) = p^k_{XY}(x_i, y_i) \cap |\mathcal{P}_{k_1}| \neq |\mathcal{P}_{k_2}|))$

Here, 2. can be simulated by adding varying levels of noise to partitions. In computer vision, these transformations can be done on the fly, using data generators, e.g., for Gaussian noise. However, this isn't the case with NLP. Due to

the grammatical structure of the text, a new carefully augmented dataset must be generated beforehand. 3. pertains to vertical Federated Learning. Similarly to most existing studies, we assume that all clients have the same ground truth, so 4. is irrelevant. Thus, we consider 5. quantity skew and 1. label distribution skew as unbalanced and non-IID distribution cases. To simulate our unbalanced and non-IID distribution cases, we partition our real-world dataset across clients using data partitioning strategies. Many existing studies use the partitioning approach to simulate federated datasets (Karimireddy et al. 2019; McMahan et al. 2017; Wang et al. 2020b). To this end, we use the following data partitioning strategies:

Quantity Skew We use a Dirichlet distribution to allocate varying amounts of data to each client. We sample $x \sim Dir_K(\beta)$, and allocate a x_k proportion of total data samples to $\mathcal{P}_k$. We fix the distribution parameter, $\beta = 10$, to control the degree of imbalance and constrain our scope. The smaller β is, the more imbalanced the distribution. Such a strategy is used in existing studies to simulate a Non-IID setting (Li et al. 2021).

Label Distribution Skew: Distribution-based imbalance Each client is allocated a proportion of the data samples of each label according to a Dirichlet distribution. We sample $p_y \sim Dir_K(\beta)$, where $\beta = 10$, and allocate a $p_{y,k}$ proportion of the samples labelled y to client k. This partitioning strategy has been used to simulate real world data distributions in recent studies (Li et al. 2021; Yurochkin et al. 2019; Wang et al. 2020a).

Label Distribution Skew: Quantity-based imbalance (u) Each client owns data samples of a fixed subset of the total v labels. Such a setting is also used in other studies (Geyer, Klein, and Nabi 2017; Li et al. 2020; Yu et al. 2020). We use a general partitioning strategy to set the number of labels, u, that each client has, where $u < v$. In the context of our study, u is either 1 or 2. The data samples of each label are split evenly across the clients that have the given label.

During the Federated Learning, we evaluate the convergence behaviour of the global model. In addition, we evaluate the global model's classification metrics, in terms of macro-average Precision, Recall and F1-score; we compare these against those of siloed, local models, as if the clients were not to participate in our system. As shown in Figure 1, there is a class-imbalance in the dataset; the hate speech class and neither class are most important to get right, and they are under-represented. The offensive class is over-represented. Therefore, macro average is preferable over weighted average, which is misleading in this case (Grandini, Bagli, and Visani 2020). F1-score is used rather than AUROC because AUROC averages over all possible discrimination thresholds, which is misleading in the case of class-imbalanced data (Chicco and Jurman 2020). In contrast, F1-score maintains a balance between Precision and Recall (Grandini, Bagli, and Visani 2020). With regards to statistical control, there are 2 assignable, random variables in our experiments: the train/test split, and the synthetic data distribution. Thus, we synthesize 10 samples using each data partitioning strategy. Additionally, we use 10-fold Cross-Validation to randomize the train/test split. Consequently, we perform 100 replicates of the experiment per data partitioning strategy and calculate the 95% Confidence Intervals.

Results

Robustness to Extreme Non-IID Setting

For quantity-based label imbalance, the data is partitioned to create two extreme cases of the Non-IID setting: Label Imbalance 1, where each client's dataset contains data samples of a single label; and Label Imbalance 2, where each client's dataset contains data samples of two distinct labels. We find that the $FedAvg$ approach that our system adopts is not robust enough to withstand such extreme cases. Firstly, in the case of Label Imbalance 1, clients are presented with the one-class classification problem (Khan and Madden 2014); this is much more challenging than the multi-class classification problem. The log loss function is only defined for two or more labels (Bishop and Nasrabadi 2006). Therefore, clients can't train their local models as they can't define a decision boundary without support by the presence of data samples from at least a positive and negative class (Khan and Madden 2014). Secondly, in the case of Label Imbalance 2, $FedAvg$ approaches used in existing studies (Li et al. 2021) withstand such a case (where $1 < u < v$) by abusing the $FedAvg$ algorithm. For example, consider the clients k_1 and k_2, where $p_Y^{k_1}(2) = 0$ and $p_Y^{k_2}(1) = 0$. After training the local models w^1 and w^2, they address different binary classification problems. The global model w is aggregated, using $FedAvg$, by directly combining the models for different classification problems, as if they addressed the same problem. The global model is subsequently evaluated on different classification problems for both clients (which are also different from our original classification problem). This makes it difficult to ascertain the extent to which the system addresses our original classification problem. In the light of this, our system clearly defines the hypothesis space in which it is to search, by stipulating that, during the evaluation round, the global model must predict probabilities for all 3 labels, on all clients. As a result, the system does not withstand this case.

To improve the robustness of our system to withstand extreme class distributions, we utilize a data sharing strategy (Zhao et al. 2018). In extreme cases of the Non-IID setting, our system partitions the train split into a client part D, to be partitioned across the clients, and a holdout part G; where $\beta = 2.5\%$, given $\beta = \frac{|G|}{n_{train}} \times 100$. Thereafter, a warm-up model is trained on G and used to initialize the local model on each client. Furthermore, we take a random α portion of G to be globally shared data that is merged with the data of each client, where $\alpha = 10\%$. Figure 14(a) shows the merged local train splits across clients. In contrast, the local test splits, shown by Figure 14(b), still have the same probability distributions as the clients' local train splits had before merging. Thus, by distributing only 0.25% of our total training data as well as the warm-up model, at the initial-

ization stage of federated learning, the system can perform in cases of the extreme Non-IID setting. Thus, results in extreme cases of the Non-IID setting assume the usage of the aforementioned data partitoning strategy. Note that the data sharing technique does not change the $FedAvg$ algorithm used in our system's approach, rather it is an additional initialization strategy added to the approach, improving the robustness of the algorithm in the extreme Non-IID setting.

Convergence Behaviour

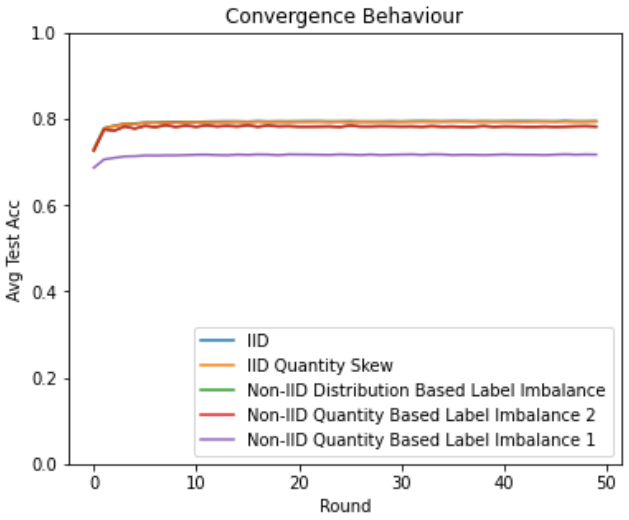


Figure 3: The average training curve of FedAvg on different Non-IID Settings

Figure 3 shows the accuracy of our system under different unbalanced and Non-IID data distribution cases. The results for the IID setting are presented as a bench mark. The accuracy does not start at the origin because each evaluation round proceeds a training round in federated evaluation. Therefore, we only evaluate the global model after the first global batch. We constrain the number of local epochs to 30, by dividing the maximum number of iterations that L-BFGS may take to converge on the full dataset by the number of federated learning rounds. This allows the convergence behaviour of our system to be observed. L-BFGS approximates Newton's method in one dimension, by computing a positive-definite finite difference approximation of the gradient (Liu and Nocedal 1989). If the objective function is non-convex, Newton's method and its approximations aren't guaranteed to converge (Weinberger 2018). If it converges, L-BFGS has at best a super-linear rate of convergence (Tibshirani 2019), given that the twice continuous differentiable assumption on the objective function holds (Chen 1996). Our log loss objective function is convex and smooth (Berrada, Zisserman, and Kumar 2018). Thus, we should see a convergence slightly better/similar to linear. In comparison, Gradient Descent has linear convergence (Tibshirani 2019). The accuracy after the fist training round is noticeably high. This is because we use a higher number of local epochs than some studies; (Zhao et al. 2018) evaluates federated Gradient Descent with local epochs of 1 and 5. This also explains why our training curve is much flatter. Our system's approach converges in at most 21 federated learning rounds. However, that's equal to 630 solver iterations, which is comparable to federated systems evaluated in other studies (Zhao et al. 2018).

From Figure 3 we can observe that, except for the case where each client only has samples of a single class, the $FedAvg$ approach has similar convergence behaviour in the Non-IID setting as in the IID setting. Firstly, for quantity skew, the accuracy is indistinguishable from that for the IID setting. This is down to the fact that $FedAvg$ weighs each client's update with the relative proportion of the client's local dataset. Secondly, in the case of label distribution skew, the algorithm's accuracy for distribution-based label imbalance is similar to that for the IID setting. The Dirichlet distribution parameter, otherwise known as the concentration parameter, parameterizes the label distribution skew. Values of the concentration parameter above 1 lead to variates that are more dense and evenly distributed, i.e., all local datasets have similar label distributions. In contrast, values of the concentration parameter below 1 lead to variates that are more sparse, i.e., all local datasets have very different label distributions, each skewed towards particular labels. Thus, by fixing $\beta = 10$, we simulate a mild case of the Non-IID setting, hence similar accuracy to that for the IID setting. In the case where clients have samples of two classes, the accuracy is comparable to that for the IID setting; although it varies more across federated learning rounds. As shown by (Zhao et al. 2018), the data sharing strategy improves accuracy in the Non-IID setting, hence the comparable accuracy. On the other-hand, the algorithm has the worst accuracy when each client only has samples of a single class.

Classification Performance Analysis

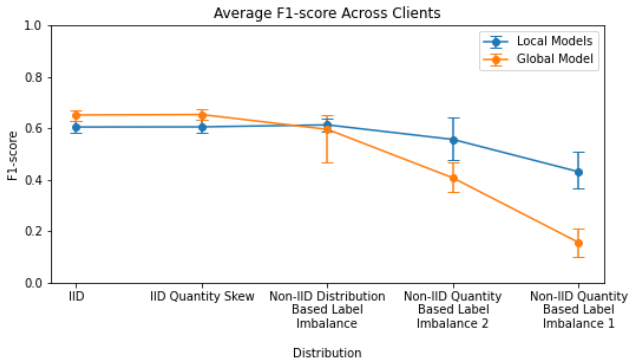


Figure 4: Average F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

Figure 4 shows the average F1-score of the ultimate global model across clients in comparison to that of local models trained on siloed clients. We calculate the average F1-score of the ultimate global model across clients in the last evaluation round, by evaluating the macro-average F1-score of the

global model on each client's local test split and taking the average of the macro-averages. Additionally, the average F1-score of local models trained on siloed clients is calculated by training each client's local model in isolation, on its respective train split, before evaluating its macro-average F1-score, on its respective test split, and taking the average of the macro-averages. It's evident that there's a negative correlation between the degree of skew in the data distribution and both the performance of the global model, and that of siloed, local models.

In the case of the IID setting and Quantity Skew, the global model performs slightly better and no worse than siloed, local models; with the average F1-score across replicates being only 7% higher relative to that of siloed, local models. In this case, an explanation for the performance of our system might be had if we view client updates in terms of mini-batches and the process of $FedAvg$ as resembling ensemble methods. Each client trains on a subset of the federated dataset before sending its updated model parameters to the server. Hence, each client essentially performs a mini-batch. Further, the aggregation performed by the server is similar to the voting process in ensemble learning. With this new perspective, the statistical problem that ensemble learning partly helps to overcome is that which occurs when an algorithm searches through a hypothesis space that is too large relative to the available training data (Dietterich and others 2002). As shown by Figure 13(a), in certain cases of the Quantity Skewed setting, clients may have less than 1k training samples. Thus, as in ensemble learning, a vote between clients reduces the risk that the hypothesis of a given client's model doesn't generalize well to future data (Dietterich and others 2002). Thereby, the performance of our global model across clients is slightly better and no worse than that of siloed, local models.

Nevertheless, the more skewed the distribution of labels, the better the siloed, local models perform relative to the global model. Whereas the global model performs similarly to the siloed, local models for Distribution-based Label Imbalance, the siloed, local models perform 27% better relative to the global model for Quantity-based label imbalance 2; the relative performance gap increases to 64% for Quantity-based label imbalance 1. Consequently, **we find that the closer the data distribution is to the IID setting, the better the performance of our global model and thus the more beneficial it would be for organisations to participate in our system**. This finding is supported by existing literature. (Ghadimi and Lan 2013) state the significance of the data distribution being as close to the IID setting as possible, such that the stochastic gradient is an unbiased estimate of the full gradient. In our context, this translates as: such that the positive-definite matrix computed at a global epoch e by a given client is an unbiased estimate of that computed by L-BFGS for epoch e on the full dataset.

To understand why our system's performance decreases as the data distribution becomes more skewed, we take a closer look at the dispersion of our global model's performance across clients. Figure 5 shows the average dispersion of the F1-score of the ultimate global model across clients in comparison to that of local models trained on siloed clients. There is a positive correlation between the degree of skew in the data distribution and both the dispersion of the performance of the global model, and that of local, siloed models. We can empirically observe that in the case of label distribution skew, the dispersion in the global model's performance across clients is generally greater relative to the dispersion in the performance of clients' siloed, local models. Figure 14 illustrates the performance of our system, given a specific data distribution synthesized with the quantity-based label imbalance 2 partitioning strategy. Evidently, each client has a very different local data distribution from the population distribution. Consequently, the local model of each client has a differing objective. Therefore, their local optima will not be the same as the global optima. In other words, in each training round, the local models take steps towards their local optima, which can be arbitrarily far away from the global optima. Over many federated learning rounds, the optima to which the global model converges will be closer to some local models than others. This is referred to as weight divergence (Zhao et al. 2018). As shown by (Zhao et al. 2018), weight divergence increases as the data distribution becomes more skewed, which explains the trend in the dispersion of the performance of the global model across clients.

Given that $FedAvg$ weighs client updates based on the relative proportion of their local datasets, clients with relatively large local datasets contribute more to the weights of the global model, as per the definition of weighted average. Figure 14 and Figure 15 both illustrate the positive correlation between the relative proportion of a client's local dataset and the performance of the global model on the client's test data. Nevertheless, this trend is not apparent in data distribution settings closer to the IID setting, such as those illustrated in Figure 11 and Figure 13. Therefore, the trend can be attributed to being a manifestation of weight divergence, where the global optima drifts closer to the local optima of clients's whose updates carry the most weight. In this vein, (Zhao et al. 2018) prove that weight divergence is bounded by the Earth Mover's Distance (EMD) between each client's class distribution and the population's class distribution. This explains why we do not see the trend between the relative proportion of a client's local dataset and the performance of the global model on the client's test data in data distributions closer to the IID setting; the weight divergence is constrained by the lower EMD.

Discussion and Conclusion

From our findings, it's evident that our system is particularly beneficial when the data distribution across clients is closer to the IID setting. Wherein, it functions similarly to ensemble methods. On the other-hand, it's usefulness decreases as the class distribution becomes more skewed, as a result of weight divergence, which (Zhao et al. 2018) show can be quantified by the EMD. Extremely skewed class distributions pose a unique set of challenges that can be addressed by clearly defining the hypothesis space in which the system is to search and using a data sharing strategy (Zhao et al. 2018) so that all clients are solving the same problem. Federated transfer learning is something different and is not the

focus of this study (Kairouz et al. 2021).

Naively aggregating client updates implicitly advantages or disadvantages some of the devices, as the global model becomes biased towards clients with a larger relative proportion of data. Recent works have proposed modified approaches to $FedAvg$ that aim to reduce the dispersion of the global model's performance across clients, such as $FedOpt$ and $FedProx$ (Li et al. 2021). Thus, this work can be extended with the introduction of such approaches.

References

[Ahmed, Vidgen, and Hale 2022] Ahmed, Z.; Vidgen, B.; and Hale, S. A. 2022. Tackling racial bias in automated online hate detection: Towards fair and accurate detection of hateful users with geometric deep learning. *EPJ Data Science* 11(1):8.

[Aluru et al. 2020] Aluru, S. S.; Mathew, B.; Saha, P.; and Mukherjee, A. 2020. Deep learning models for multilingual hate speech detection. *arXiv preprint arXiv:2004.06465*.

[Anon 2020] Anon. 2020. Understanding abusive speech detection through dataset-driven measurements.

[Aroyehun and Gelbukh 2018] Aroyehun, S. T., and Gelbukh, A. 2018. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 90–97.

[Ash and Adams 2020] Ash, J., and Adams, R. P. 2020. On warm-starting neural network training. *Advances in Neural Information Processing Systems* 33:3884–3894.

[Badjatiya et al. 2017] Badjatiya, P.; Gupta, S.; Gupta, M.; and Varma, V. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion*, 759–760.

[Bauer et al. 1994] Bauer, M. A.; Coburn, N.; Erickson, D. L.; Finnigan, P. J.; Hong, J. W.; Larson, P.-A.; Pachl, J.; Slonim, J.; Taylor, D. J.; and Teorey, T. J. 1994. A distributed system architecture for a distributed application environment. *IBM Systems Journal* 33(3):399–425.

[Berrada, Zisserman, and Kumar 2018] Berrada, L.; Zisserman, A.; and Kumar, M. P. 2018. Smooth loss functions for deep top-k classification. *arXiv preprint arXiv:1802.07595*.

[Beutel et al. 2020] Beutel, D. J.; Topal, T.; Mathur, A.; Qiu, X.; Parcollet, T.; de Gusmão, P. P.; and Lane, N. D. 2020. Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.

[Bioglio and Pensa 2022] Bioglio, L., and Pensa, R. G. 2022. Analysis and classification of privacy-sensitive content in social media posts. *EPJ Data Science* 11(1):12.

[Bishop and Nasrabadi 2006] Bishop, C. M., and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

[Brownlee 2016] Brownlee, J. 2016. *Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end*. Machine Learning Mastery.

[Brownlee 2019] Brownlee, J. 2019. *Probability for machine learning: Discover how to harness uncertainty with Python*. Machine Learning Mastery.

[Burnap and Williams 2016] Burnap, P., and Williams, M. L. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data science* 5:1–15.

[Chen 1996] Chen, X. 1996. Convergence of the bfgs method for lcˆ1 convex constrained optimization. *SIAM Journal on Control and Optimization* 34(6):2051–2063.

[Chicco and Jurman 2020] Chicco, D., and Jurman, G. 2020. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics* 21(1):1–13.

[Davidson et al. 2017] Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, 512–515.

[Dean et al. 2012] Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Ranzato, M.; Senior, A.; Tucker, P.; Yang, K.; et al. 2012. Large scale distributed deep networks. *Advances in neural information processing systems* 25.

[Dietterich and others 2002] Dietterich, T. G., et al. 2002. Ensemble learning. *The handbook of brain theory and neural networks* 2(1):110–125.

[Fan et al. 2008] Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; and Lin, C.-J. 2008. Liblinear: A library for large linear classification. *the Journal of machine Learning research* 9:1871–1874.

[Fortuna and Nunes 2018] Fortuna, P., and Nunes, S. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51(4):1–30.

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

[Ghadimi and Lan 2013] Ghadimi, S., and Lan, G. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23(4):2341–2368.

[Graham 2012] Graham, C. 2012. Anonymisation: managing data protection risk code of practice. *Information Commissioner's Office*.

[Grandini, Bagli, and Visani 2020] Grandini, M.; Bagli, E.; and Visani, G. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

[Kairouz et al. 2021] Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning* 14(1–2):1–210.

[Karim et al. 2021] Karim, M. R.; Dey, S. K.; Islam, T.; Sarker, S.; Menon, M. H.; Hossain, K.; Hossain, M. A.; and Decker, S. 2021. Deephateexplainer: Explainable hate speech detection in under-resourced bengali language. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, 1–10. IEEE.

[Karimireddy et al. 2019] Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S. J.; Stich, S. U.; and Suresh, A. T. 2019. Scaffold: Stochastic controlled averaging for on-device federated learning.

[Kelley 1995] Kelley, C. T. 1995. *Iterative methods for linear and nonlinear equations*. SIAM.

[Khan and Madden 2014] Khan, S. S., and Madden, M. G. 2014. One-class classification: taxonomy of study and re-

view of techniques. *The Knowledge Engineering Review* 29(3):345–374.

[Koops 2014] Koops, B.-J. 2014. The trouble with european data protection law. *International data privacy law* 4(4):250–261.

[Kumar et al. 2018] Kumar, R.; Ojha, A. K.; Malmasi, S.; and Zampieri, M. 2018. Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018)*, 1–11.

[Li et al. 2020] Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems* 2:429–450.

[Li et al. 2021] Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2021. Federated learning on non-iid data silos: An experimental study. *arXiv preprint arXiv:2102.02079*.

[Liu and Miller 2020] Liu, D., and Miller, T. 2020. Federated pretraining and fine tuning of bert using clinical notes from multiple silos. *arXiv preprint arXiv:2002.08562*.

[Liu and Nocedal 1989] Liu, D. C., and Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming* 45(1):503–528.

[Lucas 2014] Lucas, B. 2014. Methods for monitoring and mapping online hate speech. *GSDRC Applied Knowledge Services* 14.

[McMahan et al. 2017] McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

[Murphy 2022] Murphy, M. H. 2022. Assessing the implications of schrems ii for eu–us data flow. *International & Comparative Law Quarterly* 71(1):245–262.

[Party 2007] Party, D. P. W. 2007. Opinion 4/2007 on the concept of personal data. *Brussels, Belgium: European Commission*.

[Pedregosa et al. 2011] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research* 12:2825–2830.

[Pelicon, Martinc, and Novak 2019] Pelicon, A.; Martinc, M.; and Novak, P. K. 2019. Embeddia at semeval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 604–610.

[Pereira-Kohatsu et al. 2019] Pereira-Kohatsu, J. C.; Quijano-Sánchez, L.; Liberatore, F.; and Camacho-Collados, M. 2019. Detecting and monitoring hate speech in twitter. *Sensors* 19(21):4654.

[Poletto et al. 2021] Poletto, F.; Basile, V.; Sanguinetti, M.; Bosco, C.; and Patti, V. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation* 55(2):477–523.

[Reddi et al. 2020] Reddi, S.; Charles, Z.; Zaheer, M.; Garrett, Z.; Rush, K.; Konečnỳ, J.; Kumar, S.; and McMahan, H. B. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

[Ribeiro et al. 2017] Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A.; and Meira Jr, W. 2017. " like sheep among wolves": Characterizing hateful users on twitter. *arXiv preprint arXiv:1801.00317*.

[Ribeiro et al. 2018] Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A.; and Meira Jr, W. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*.

[Robinson et al. 2009] Robinson, N.; Graux, H.; Botterman, M.; and Valeri, L. 2009. Review of the european data protection directive. *Rand Europe*.

[Samonte 2020] Samonte, M. 2020. Google v. cnil: The territorial scope of the right to be forgotten under eu law. European Papers (European Forum Insight/Highlight).

[Sartor and Lagioia 2020] Sartor, G., and Lagioia, F. 2020. The impact of the general data protection regulation (gdpr) on artificial intelligence. Technical report, Directorate-General for Parliamentary Research Services (EPRS) of the Secretariat of the European Parliamen.

[Schmidt and Wiegand 2019] Schmidt, A., and Wiegand, M. 2019. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017, Valencia, Spain*, 1–10. Association for Computational Linguistics.

[Śmietanka, Pithadia, and Treleaven 2020] Śmietanka, M.; Pithadia, H.; and Treleaven, P. 2020. Federated learning for privacy-preserving data access. *Available at SSRN 3696609*.

[Stalla-Bourdillon and Knight 2016] Stalla-Bourdillon, S., and Knight, A. 2016. Anonymous data v. personal data-false debate: an eu perspective on anonymization, pseudonymization and personal data. *Wis. Int'l LJ* 34:284.

[Tibshirani 2019] Tibshirani, R. 2019. Newton's method. In *Notes for Convex Optimization: Machine Learning 10-725*.

[Wang et al. 2020a] Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020a. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.

[Wang et al. 2020b] Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020b. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems* 33:7611–7623.

[Waseem and Hovy 2016] Waseem, Z., and Hovy, D. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, 88–93.

[Weinberger 2018] Weinberger, K. 2018. Gradient descent (and beyond). Cornell CS4780 SP17.

[Wulczyn, Thain, and Dixon 2017] Wulczyn, E.; Thain, N.; and Dixon, L. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, 1391–1399.

[Yang et al. 2019] Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10(2):1–19.

[Yu et al. 2020] Yu, F.; Rawat, A. S.; Menon, A.; and Kumar, S. 2020. Federated learning with only positive labels. In *International Conference on Machine Learning*, 10946–10956. PMLR.

[Yurochkin et al. 2019] Yurochkin, M.; Agarwal, M.; Ghosh, S.; Greenewald, K.; Hoang, N.; and Khazaeni, Y. 2019. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, 7252–7261. PMLR.

[Zhao et al. 2018] Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*.

Appendix

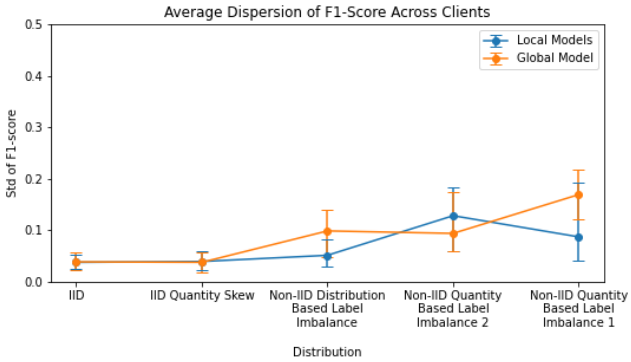


Figure 5: Average dispersion of F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

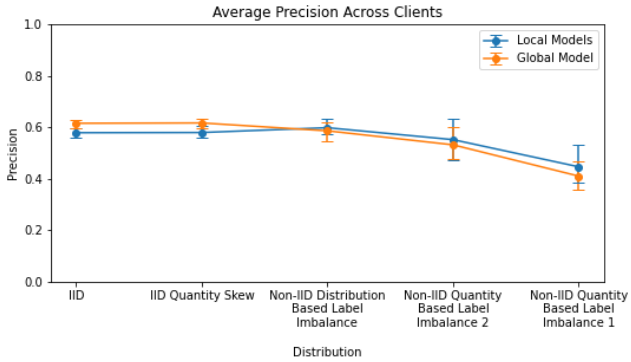


Figure 6: Average precision of ultimate global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

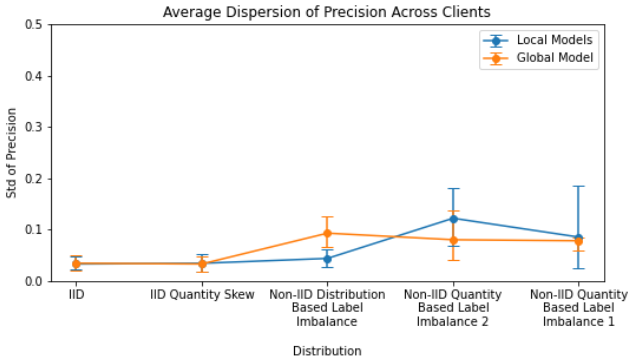


Figure 7: Average dispersion of ultimate precision of global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

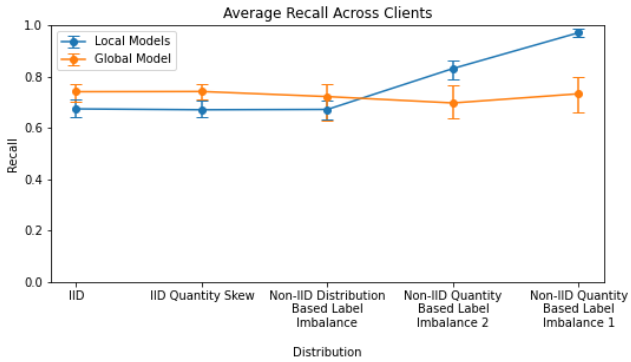


Figure 8: Average recall of ultimate global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

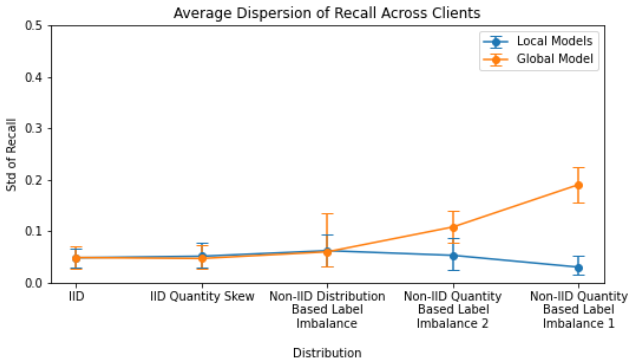
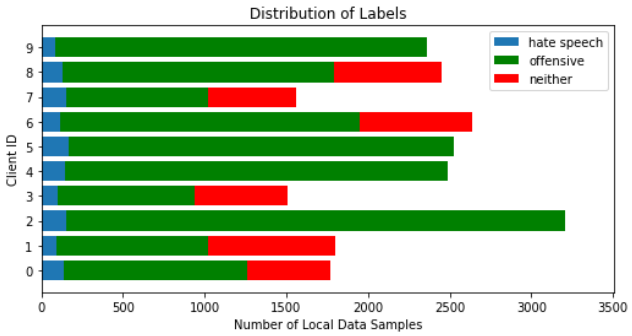


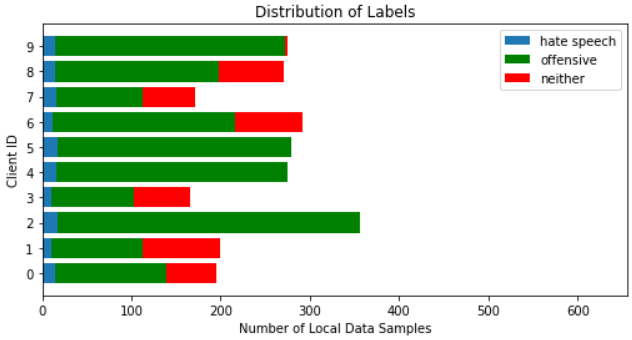
Figure 9: Average dispersion of recall of ultimate global model across clients in comparison to that of local models trained on siloed clients (95% CI's)

Distribution	Precision (95% CI)		Recall (95% CI)	
	Global	**Local**	**Global**	**Local**
IID	0.615 (0.596-0.630)	0.579 (0.560-0.597)	0.741 (0.701-0.772)	0.674 (0.640-0.711)
IID Quantity Skew	0.617 (0.598-0.634)	0.579 (0.559-0.605)	0.742 (0.711-0.771)	0.671 (0.643-0.707)
Non-IID Distribution Based Label Imbalance	0.586 (0.545-0.620)	0.598 (0.574-0.632)	0.722 (0.628-0.770)	0.672 (0.635-0.708)
Non-IID Quantity Based Label Imbalance 2	0.531 (0.477-0.601)	0.551 (0.474-0.633)	0.697 (0.640-0.766)	0.832 (0.787-0.861)
Non-IID Quantity Based Label Imbalance 1	0.410 (0.358-0.467)	0.446 (0.383-0.533)	0.733 (0.659-0.800)	0.971 (0.956-0.985)

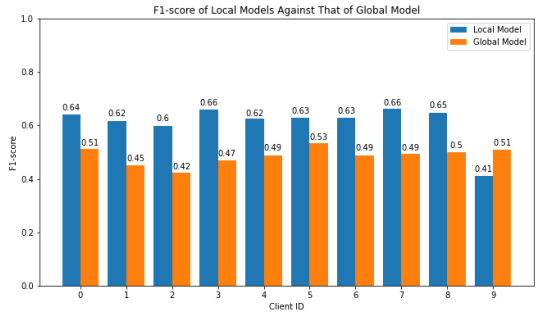
Figure 10: Average precision and recall of ultimate global model across clients in comparison to that of local models trained on siloed clients



(a) Partitioned train split

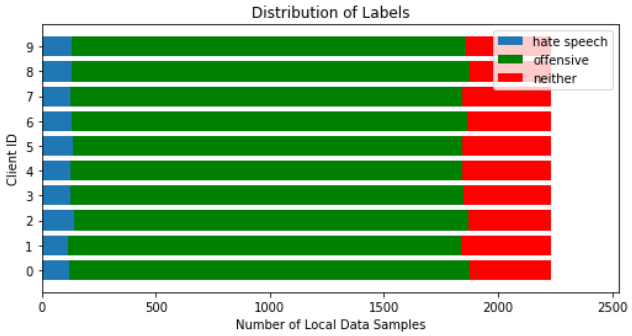


(b) Partitioned test split

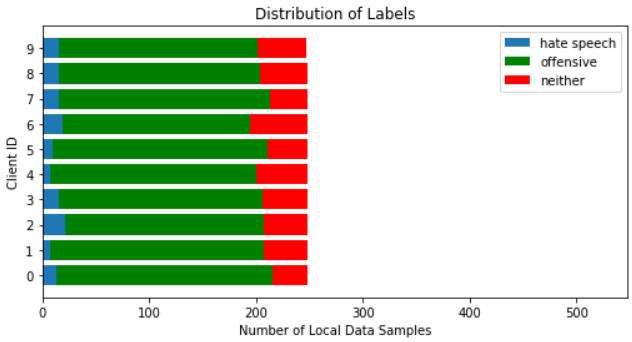


(c) F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients

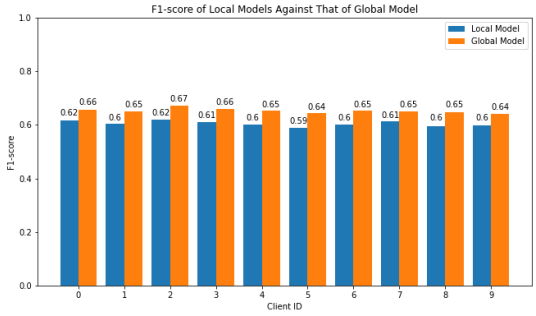
Figure 11: Performance in particular data distribution synthesized with the distribution-based label imbalance partitioning strategy



(a) Partitioned train split

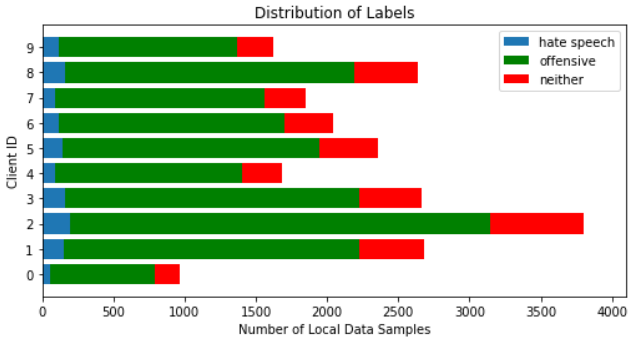


(b) Partitioned test split

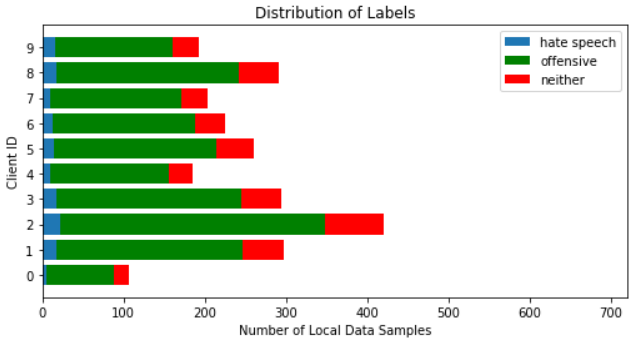


(c) F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients

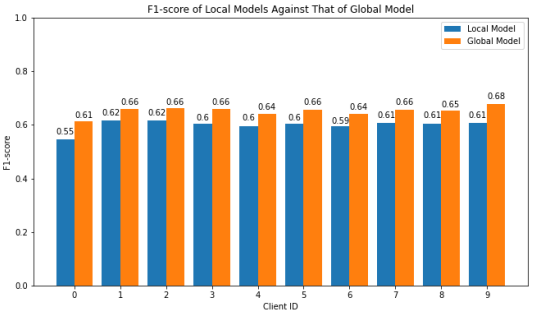
Figure 12: Performance in particular data distribution of the IID setting



(a) Partitioned train split

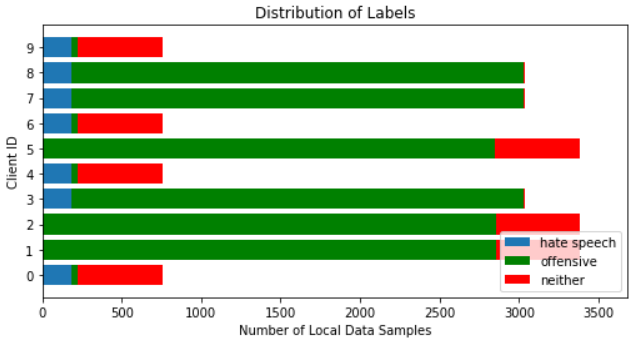


(b) Partitioned test split

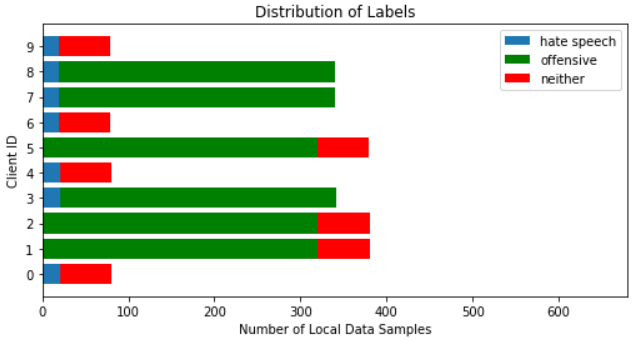


(c) F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients

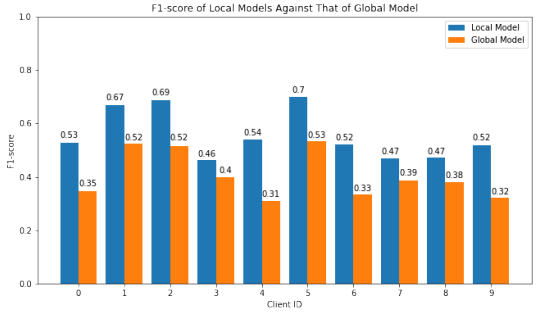
Figure 13: Performance in particular data distribution synthesized with the quantity skew partitioning strategy



(a) Partitioned train split

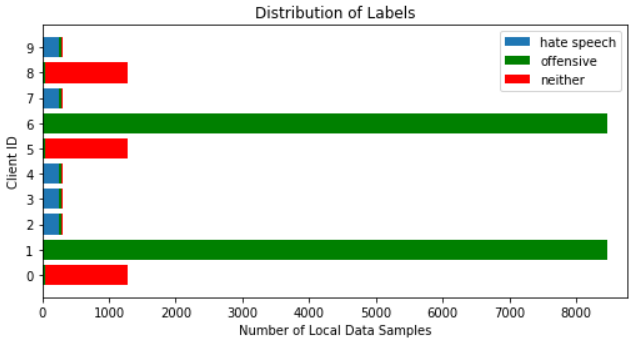


(b) Partitioned test split

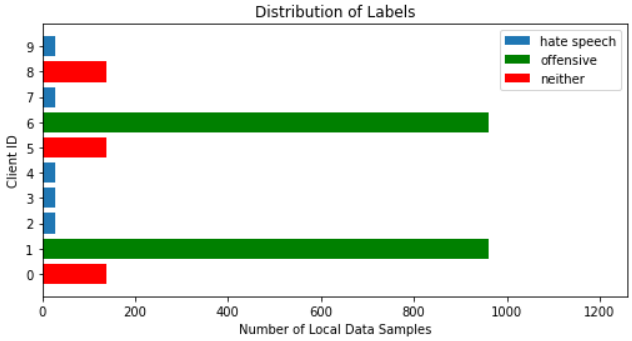


(c) F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients

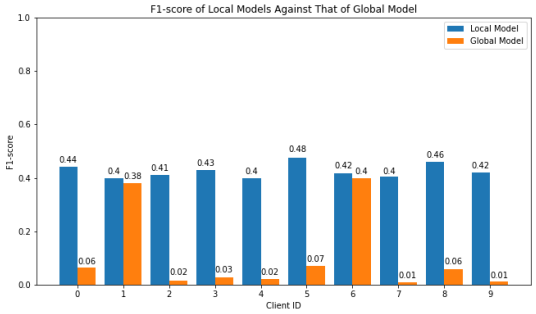
Figure 14: Performance in particular data distribution synthesized with the quantity-based label imbalance 2 partitioning strategy



(a) Partitioned train split



(b) Partitioned test split



(c) F1-score of ultimate global model across clients in comparison to that of local models trained on siloed clients

Figure 15: Performance in particular data distribution synthesized with the quantity-based label imbalance 1 partitioning strategy

Distribution	Std of Precision Across Clients (95% CI)		Std of Recall Across Clients (95% CI)	
	Global	**Local**	**Global**	**Local**
IID	0.034 (0.019-0.050)	0.033 (0.022-0.048)	0.048 (0.027-0.069)	0.048 (0.030-0.067)
IID Quantity Skew	0.033 (0.016-0.048)	0.034 (0.018-0.051)	0.047 (0.027-0.073)	0.051 (0.030-0.077)
Non-IID Distribution Based Label Imbalance	0.093 (0.066-0.125)	0.043 (0.026-0.061)	0.060 (0.030-0.135)	0.062 (0.032-0.093)
Non-IID Quantity Based Label Imbalance 2	0.080 (0.040-0.136)	0.122 (0.067-0.180)	0.108 (0.077-0.138)	0.053 (0.025-0.087)
Non-IID Quantity Based Label Imbalance 1	0.078 (0.058-0.083)	0.085 (0.024-0.186)	0.190 (0.155-0.225)	0.030 (0.014-0.053)

Figure 16: Average dispersion of precision and recall of ultimate global model across clients in comparison to that of local models trained on siloed clients