Private Federated Learning with Domain Adaptation

Daniel W. Peterson, Pallika Kanani, Virendra J. Marathe, Rave Harpaz, Steve Bright

Federated learning (FL) was originally motivated by communication bottlenecks in training models from data stored across millions of devices, but the paradigm of distributed training is attractive for models built on sensitive data, even when the number of users is relatively small, such as collaborations between organizations. For example, when training machine learning models from health records, the raw data may be limited in size, too sensitive to be aggregated directly, and concerns about data reconstruction must be addressed. Differential privacy (DP) offers a guarantee about the difficulty of reconstructing individual data points, but achieving reasonable privacy guarantees on small datasets can significantly degrade model accuracy. Data heterogeneity across users may also be more pronounced with smaller numbers of users in the federation pool. We provide a theoretical argument that model personalization offers a practical way to address both of these issues, and demonstrate its effectiveness with experimental results on a variety of domains, including spam detection, named entity recognition on case narratives from the Vaccine Adverse Event Reporting System (VAERS) and image classification using the federated MNIST dataset (FEMNIST).

1 Introduction

Federated Learning (FL) is a distributed ML paradigm that enables multiple users to jointly train a shared model without sharing their data with any other users [Bonawitz et al.(2019)Bonawitz, Eichner, Grieskamp, Huba, Ingerman, Ivanov, Kiddon, Konecný, Mazzocchi, Mc Konecný, McMahan, and Ramage(2015)], offering advantages in both scale and privacy. In FL, multiple users wish to perform essentially the same task using ML, with a model architecture that is agreed upon in advance. Each user wants the best possible model for their individual use, but often has a limited budget for labeling

Oracle Labs
{daniel.peterson,pallika.kanani,virendra.marathe,rave.harpaz,steve.bright}@oracle.com

their own data. Pooling the data of multiple users could improve model accuracy, because accuracy generally increases with increased training data. However user data cannot be shipped to a common model training facility due to bandwidth limitations or data privacy concerns. As a result, users locally train the shared (global) model on their local data, and thereafter send the updated model to the *federation server*. The federation server aggregates updates received from its users to improve the global model for all users.

Although the initial focus of FL has been on targeting millions of mobile devices [Bonawitz et al.(2019)Bonawitz, Eichner, Grieskamp, Huba, Ingerman, Ivanov, Kiddon, Konecný, Mazzocchi, Mc
also called *cross-device FL*, the benefits of its architecture are evident even for institutional settings, also called *cross-silo FL* [Kairouz et al.(2019)Kairouz, McMahan, Avent, Bellet, Bennis, Bhagoji, Bonawit
While cross-device FL is concerned with both bandwidth consumption and data privacy, cross-silo federations and their users are considered well equipped with resources to handle bandwidth concerns, and data privacy is the primary objective. Our work focuses on the cross-silo FL setting.

Today our world grapples with safely rolling out massive scale vaccination programs to end a pandemic. Understanding adverse events related to these vaccines is critically important. These adverse events are often expressed in free text form, such as social media posts and reports provided to health care agencies and pharmaceutical companies. Currently, mentions of specific adverse events are extracted and coded manually, which is a time consuming, expensive and non-scalable process. Therefore, Machine Learning (ML) based methods to extract named entities (adverse events) automatically from such unstructured data are highly desirable.

Typically, more training data yield more accurate models. Unfortunately, collecting human annotations for building such Named Entity Recognition (NER) models is expensive, and particularly challenging given the need to maintain privacy of health records. One way to overcome this data scarcity issue would be for various agencies to share their data to build a joint model with combined data. However, privacy concerns, government regulation and data use agreements might not allow the data to leave individual organizational or geographical silos. Sharing user data with other users is absolutely not an option in these settings.

Cross-silo FL makes perfect sense to address such problems. Each vaccine provider's data remains in its private *silo*. At the same time, the provider can collaborate with other providers on a FL framework to collectively improve the NER model used for adverse event detection. Everyone benefits without violating data privacy. More specifically, for institutions participating in a federation as users, restricting data movement helps fulfill contractual obligations with their customers and comply with legal regulatory constraints on data movement [ccpa(), gdpr()].

However, restricting the provider's training data to its private silo does not guarantee complete privacy. Recent works have demonstrated that the data can indirectly leak out through model updates shipped by users to the federation server [Bagdasaryan et al.(2020)Bagdasaryan, Veit, Hua, Estrin, and Shmatikov, Melis et al.(2018)Melis, Song, Cristofar
Nasr, Shokri, and Houmansadr(2019)]. To combat this problem, researchers have proposed the addition of Differential Privacy (DP) [Dwork(2006), Dwork and Roth(2014),
Dwork et al.(2006)Dwork, McSherry, Nissim, and Smith] to FL [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan,

Geyer, Klein, and Nabi(2017), Konecný et al.(2016)Konecný, McMahan, Ramage, and Richtárik, McMahan et al.(2017)McMahan, Ramage, Talwar, and Zhang].

Informally, DP aims to provide a bound on the variation in the model's output based on the inclusion or exclusion of a single data point used in its training set. This is done by introducing precisely calibrated noise in the training process. The method of noise calibration and injection varies between implementations [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang, McMahan et al.(2017)McMahan, Ramage, Talwar, and Zhang], but is always structured to enforce the precise formal DP guarantee, which we define in section 2. We will refer to this process as "DP inducing noise injection" henceforth. This noise makes it difficult, even impossible, to determine whether any particular data point was used to train the model. While this noise is structured to enforce formally provable privacy guarantees for the data point [Dwork et al.(2006)Dwork, McSherry, Nissim, and Smith], it can degrade accuracy of model predictions. Figure 1 depicts performance of an FL model that enforces different levels of DP guarantees for training data (No Privacy, $\varepsilon = [2, 4]$). Clearly, as the DP related bounds grow tighter (smaller value of ε), indicating better privacy guarantees, the FL model delivers worse performance, despite the larger training dataset available through FL. In settings where the federation server is trusted, DP enforcement is delegated to the federation server [McMahan et al.(2017)McMahan, Ramage, Talwar, and Zhang]. However, in settings where users do not trust even the federation server, DP may need to be enforced by the users locally [Kasiviswanathan et al.(2008)Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith]. While all this noise is structured to enforce formally provable privacy guarantees for each training data point [Dwork et al.(2006)Dwork, McSherry, Nissim, and Smith], it can significantly degrade accuracy of model predictions. This degradation may happen to an extent that disincentivizes users from participating in the federation – the global (noisy) model performs worse than a user-resident local model trained just on the user's dataset, which we call the *individual* model.

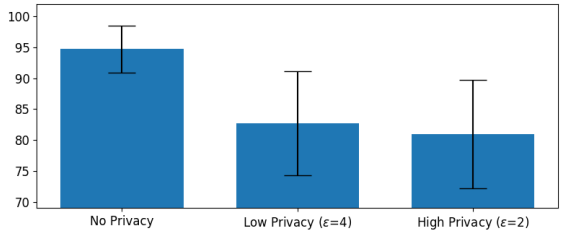

Fig. 1 Classifier accuracy (higher is better) on a spam classification dataset comprising 15 users cooperating in a FL setting (more details on the benchmark in Evaluation section). Introduction of DP-induced noise significantly compromises accuracy.

Another instance where the global model may perform worse than the individual model for a user is when the user's data distribution is different from most of the users, or the users collectively have non-IID training data [Hsieh et al.(2019)Hsieh, Phanishayee, Mutlu, and Gibbons,

Li et al.(2020b)Li, Huang, Yang, Wang, and Zhang]. There is a rapidly growing body of FL *Personalization* literature to address this problem [Dinh, Tran, and Nguyen(2020), Fallah, Mokhtari, and Ozdaglar(2020), Liang et al.(2020)Liang, Liu, Liu, Salakhutdinov, and Morency, Mansour et al.(2020)Mansour, Mohri, Ro, and Suresh, Peterson, Kanani, and Marathe(2019), Yu, Bagdasaryan, and Shmatikov(2020)], a handful of which addresses model degradation due to DP induced noise [Peterson, Kanani, and Marathe(2019), Yu, Bagdasaryan, and Shmatikov(2020)].

We are interested in applying this body of work to real-world problem settings. The health care sector is one such application domain that can leverage FL in significant ways. Indeed there is rapidly growing awareness and investment in FL at world-wide scale including consortiums [mellody()] and public-private partnerships [imi()]. This is accompanied by the beginnings of applied research in this sector [Li et al.(2020a)Li, Gu, Dvornek, Staib, Ventola, and Duncan].

In this paper, we study application of cross-silo FL to various problems, ranging from spam classification to vaccine adverse event detection. The commonalities across our settings are that the siloed data comes from diverse domains.

There exists a large body of work on domain adaptation in non-FL systems [Ben-David et al.(2010)Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan, Crammer, Kearns, and Wortman(2008), Kouw and Loog(2018), Pan and Yang(2010), Daumé III(2009)]. In domain adaptation, a model trained over a dataset from a source domain is further refined to adapt to a dataset from a different target domain. We hypothesize that along with bridging the data distribution gap, domain adaptation can also address the aforementioned problem of accuracy reduction in differentially private FL. In this work, we use domain adaptation techniques to personalize the trained model to the individual users, and demonstrate empirically that personalization of a jointly-trained model can improve performance over individual training accuracy, overcoming the main drawback of differential privacy.

One technique we propose is a Mixture of Experts (MoE), where outputs of a number of domain-expert models are combined to derive the refined output. More specifically, we propose a framework to augment the FL setting with per-user domain adaptation, which can improve accuracy for individual users. Furthermore, the improvement is much more pronounced when differential privacy bounds are imposed on the FL model.

We use differentially private FL to train a public, general model on the task. We also learn a private, domain-specific model using each user's own data. Each user combines the output of the general and private models using a mixture of experts (MoE) [Masoudnia and Ebrahimpour(2014), Nowlan and Hinton(1991)] to make their final predictions. The two "experts" in the mixture are the general FL model and the domain-tuned private model, so we refer to our system as federated learning with domain experts (FL+DE). For privacy in the general model, we use FL with differentially private stochastic gradient descent (DPSGD) [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang]. The private domain models are trained using ordinary stochastic gradient descent (i.e. without differential privacy noise). In principle, the two model architectures can be identical or radically different, but for convenience we maintain a common model architecture for the general (public) and private models. Using a MoE architecture

allows the general and private models to influence predictions differently on individual data points.

We demonstrate on synthetic as well as a real-world spam detection [Bickel(2006)] datasets that our system significantly outperforms the accuracy of differentially private federated learning (DPFL). This largely boils down to two factors. First, the private models provide domain adaptation, which is known to typically increase accuracy in each domain. On a real-world classification task, we observe that domain adaptation improves accuracy even when using non-private SGD. Second, the private models allow noise-free updates, because there is no need to conceal private data from the private model. While the accuracy of the DPFL system degrades by 12% in the low-noise setting, the performance of FL+DE degrades by <0.2%. In the high-noise setting, the accuracy of the differentially-private FL system degrades by 12.1% and FL+DE accuracy degrades by only 1.1%. Although the FL+DE system does not quite match the performance of FL in the noise-free, zero-privacy setting, it clearly outperforms ordinary FL when trained with DPSGD. We additionally analyze the implications of our MoE architecture on the shared general model's stability, and the impact of individual users on FL+DE's accuracy and stability.

We study implications of applying FL to train a Named Entity Recognition (NER) model on the Vaccine Adverse Event Reporting System (VAERS) dataset that we have annotated and partitioned by vaccine manufacturers. Each vaccine manufacturer acts as a federation user whose dataset is *siloed* in its private sandbox; all these sandboxes participate in our FL framework over multiple training rounds.

Our experiments reveal several interesting insights including general effectiveness of FL on model performance, effects of DP enforcement on model performance, and the value of personalization techniques to incentivize users to participate in FL. In particular, we show that FL improves average F1 value by 37.43% over the individual model, while enforcement of DP (DP-FL) degrades the FL model's average F1 by 25.17%. For one of the users, this degradation is so severe that the private FL model F1 is worse by 45.55% when compared with the individual model F1. This clearly makes DP-FL a non-starter for some users to join the federation. We study FL with *Fine-Tuning* (FT-FL) [Yu, Bagdasaryan, and Shmatikov(2020)], a personalization approach that fine-tunes the global model at each user *after* the entire FL training process completes. Interestingly, contrary to prior work [Yu, Bagdasaryan, and Shmatikov(2020)], simply augmenting fine-tuning to FL does not result in prediction accuracy improvement for the federation users. Instead, user accuracy degrades in most cases. However, somewhat surprisingly, fine-tuning in the presence of DP (FT-DP-FL) boosts user accuracy by 24.88%, compared to the individual model, to strongly incentivize users to join and stay with the federation. We also observe that vaccine reports related to different manufacturers have slightly different vocabulary (e.g. mentions of different vaccine names), and different distributions of adverse events, which aid FT-DP-FL in effectively recovering lost accuracy.

Even more interestingly, our findings indicate a unique *incentive structure* for users to join the federation. In particular, we find that users with small amount of training data, a.k.a. *small* users, have a strong incentive to join and stay with the federation even when DP is enforced without fine-tuning. This is because the user's

private dataset is so small that any locally trained individual model performs poorly. Furthermore, even the global model that is degraded because of DP inducing noise performs significantly better than the user's individual model. In short, small users have virtually no incentive to leave the federation, and may not require additional layers of personalization to improve the global model as long as there are enough participants in the federation.

For users with larger amount of data, the narrative is quite different. In particular, we observe that the global model's degradation due to DP inducing noise is significant enough to disincentivize those users from participating in the federation. As a result, if they opt for the additional layer of privacy through DP, the importance of personalization based enhancements, which salvage the accuracy lost due to DP inducing noise, cannot be understated.

2 Background

Federated Learning (FL)

In FL, a federation server initializes a global model and ships it to all participating users thereby initiating distributed training. Training happens over multiple rounds. In each round, each user, on receiving the global model re-trains the model on its private data and sends back the resulting parameter updates to the federation server. The federation server aggregates updates from all users applying them to the global model, and then ships the revised model back to the users. The most widely used method of aggregation is FedAvg [Konecný, McMahan, and Ramage(2015), McMahan et al.(2016)McMahan, Moore, Ramage, and y Arcas], where user parameter updates are averaged at the federation server and applied to the global model. Formally, FedAvg solves the following optimization problem:

$$\min_{w \in \mathcal{R}^d} f(w) \qquad where, f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w) \tag{1}$$

The function $f_i = \mathcal{L}(w; x_i, y_i)$ represents the local loss for each of the n federation users on the model w using the user's private data x_i, y_i.

Figure 2 shows the overall FL architecture. Users can dynamically join the federation or drop out. The framework is structured to be resilient to such changes. Noting privacy concerns, more recent work has proposed addition of differential privacy to FL [Geyer, Klein, and Nabi(2017), Konecný et al.(2016)Konecný, McMahan, Ramage, and Richtárik, McMahan et al.(2016)McMahan, Moore, Ramage, and y Arcas].

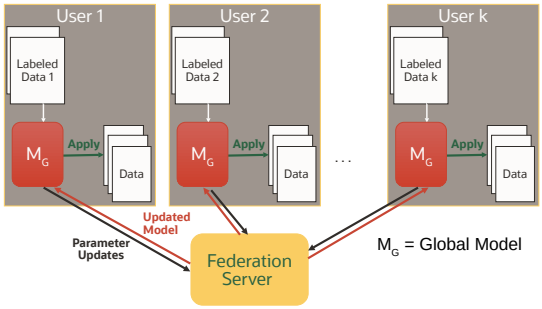

Fig. 2 The Federated Learning setting. M_G is the global model the federation server sends to users, each of which re-trains M_G on its private data and sends the updated model parameters back to the federation server.

Differential Privacy (DP)

Differential Privacy [Dwork et al.(2006)Dwork, McSherry, Nissim, and Smith] is a mathematically quantifiable privacy guarantee for a data set used by a computation that analyzes it. While it originally emerged in the database and data mining communities, triggered by privacy concerns in Machine Learning (ML) [Fredrikson, Jha, and Ristenpart(2015), Hitaj, Ateniese, and Perez-Cruz(2017), Korolova(2010), Shokri et al.(2017)Shokri, Stronati, Song, and Shmatikov, Tramèr et al.(2016)Tramèr, Zhang, Juels, Reiter, and Ristenpart], DP has garnered enormous traction in the ML community over the last decade [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Miror Carlini et al.(2019)Carlini, Liu, Erlingsson, Kos, and Song, Chaudhuri, Monteleoni, and Sarwate(2011), Differential Privacy Team(2017), Dimitrakakis et al.(2017)Dimitrakakis, Nelson, Zhang, Mitrokotsa, and Rubinstein].

In DP, the privacy guarantee applies to each individual item in the data set and is formally specified in terms of a pair of data sets that differ in at most one item. Specifically, consider an algorithm A such that $A : D \mapsto R$, where D and R are respectively the domain and range of A. Now consider two data sets d and d' that differ from each other in exactly one data item. Such data sets are considered *adjacent* to each other in the DP literature. Algorithm A is said to be (ε, δ)-differentially private if the following condition holds true for all adjacent d and d' and any subset of outputs $O \subseteq R$:

$$P[A(d) \in O] \leq e^{\varepsilon} P[A(d') \in O] + \delta \tag{2}$$

Enforcement of DP typically translates into introduction of a "correction" in algorithm A to ensure that the differential privacy bound holds for any two adjacent inputs. This correction is commonly referred to as the *noise* introduced in the algorithm, its input, or output to ensure that the (ε, δ)-differential privacy bound holds. While a disciplined introduction of noise guarantees DP, the noise itself leads

to accuracy degradation in the output produced by A. In the context of ML, the algorithm is a model being trained using sensitive private data sets, and accuracy degradation can significantly hamper the model's utility.

Personalization in FL

The basic FL algorithm, FedAvg, assumes IID training data across all FL users. In fact, it is known to be quite effective in practice for such data distributions. However, FedAvg may perform poorly in the presence of non-IID user data [Hsieh et al.(2019)Hsieh, Phanishayee, Mutlu, and Gibbons, Li et al.(2020b)Li, Huang, Yang, Wang, and Zhang]. A recent flurry of research addresses this problem using *personalization* techniques [Dinh, Tran, and Nguyen(2020), Fallah, Mokhtari, and Ozdaglar(2020), Liang et al.(2020)Liang, Liu, Liu, Salakh Mansour et al.(2020)Mansour, Mohri, Ro, and Suresh, Peterson, Kanani, and Marathe(2019), Yu, Bagdasaryan, and Shmatikov(2020)] that specialize training at each user, typically in the form of training an additional local model, or letting the local copy of the global model "drift" from the global model in a constrained fashion. This enables the local model to fit better to the user's local data distribution thereby delivering a better performing model.

Personalization techniques are related to the classic ML problem of domain adaptation. Domain adaptation is the challenge of adjusting ML models to perform on different data sets or different target tasks. A classic domain adaptation setup models a large amount of labeled data drawn from one distribution (source domain), and a pool of unlabeled or partially-labeled data drawn from another distribution (target domain). Because the domains have different distributions, a model trained only on source-domain data is unlikely to perform optimally on the target domain. Many domain-adaptation techniques have been proposed that successfully leverage labeled data in the source domain to improve model performance in the target domain [Crammer, Kearns, and Wortman(2008), Daumé and Marcu(2006), Daumé III(2009), French, Mackiewicz, and Fisher(2017), Samdani and Yih(2011), Sun and Shi(2013)].

Of particular interest to our work is the Mixture of Experts (MoE) technique used for domain adaptation [Masoudnia and Ebrahimpour(2014), Nowlan and Hinton(1991)], which bears some resemblance to other domain adaptation techniques [Guo, Shah, and Barzilay(2018), Tu and Sun(2012)]. This is one of the techniques we propose in this work.

Adverse Event Mention Extraction

By some estimates, adverse drug reactions are among the leading causes of death in the developed world. Reports of adverse events are a critical source of information for tracking and studying adverse events associated with medicinal products. However, portions of the sought information is only available in unstructured format. The use of and necessity of automated methods for extracting mentions of drug adverse events from unstructured text is widely recognized in pharmacovigilance

[Harpaz et al.(2014)Harpaz, Callahan, Tamang, Low, Odgers, Finlayson, Jung, LePendu, and Shah]. Several different genres of text are tackled in this line of research, including social media [Gurulingappa et al.(2012)Gurulingappa, Rajput, Roberts, Fluck, Hofmann-Apitius, and Toldo, Korkontzelos et al.(2016)Korkontzelos, Nikfarjam, Shardlow, Sarker, Ananiadou, and Gonzalez], biomedical literature [Leaman et al.(2010)Leaman, Wojtulewicz, Sullivan, Skariah, Yang, and Gonzalez, Winnenburg et al.(2015)Winnenburg, Sorbello, Ripple, Harpaz, Tonning, Szarfman, Francis, and Bodenreider], clinical narratives [Haerian et al.(2012)Haerian, Varn, Vaidya, Ena, Chase, and Friedman, LePendu et al.(2013)LePendu, Iyer, Bauer-Mehren, Harpaz, Mortensen, Podchiyska, Ferris, and Shah] and drug labels [Roberts, Demner-Fushman, and Tonning(2017)]. More recently, use of state of the art deep learning technology for NER have been proposed [Giorgi and Bader(2018)].

3 Federated Learning Framework

We have implemented our own FL simulation framework, on PyTorch6, that hosts the federation server and users on the same computer. The framework supports several federated aggregation protocols, including FedAvg and FedSGD [Konecný, McMahan, and Ramage(2015)], of which we use FedAvg in our evaluation. The framework is extendable to support other custom aggregation protocols [Dinh, Tran, and Nguyen(2020), Fallah, Mokhtari, and Ozdaglar(2020), Liang et al.(2020)Liang, Liu, Liu, Salakhut Peterson, Kanani, and Marathe(2019), Yu, Bagdasaryan, and Shmatikov(2020)].

3.1 Trust Model Considerations and Differential Privacy

The decision to train a ML model using the FL framework requires careful analysis of privacy considerations for users' data. More specifically, the *meaning* of the term "data privacy" in a given setting needs to be precisely understood since it has profound implications on techniques required to enforce the desired data privacy. For instance, in some settings, simply restricting user data to its private silo is sufficient for the use case. On the other hand, in settings involving highly sensitive private data (e.g. health records of individuals), it may be desirable to ensure that even the parameter updates shipped from the user silo to the federation server cannot be reverse engineered by any means, external to the user, to determine the user's training data records. Ultimately, the level of privacy protection must be agreed upon by all parties involved. While an exhaustive treatment of a taxonomy of such *trust models* in FL is beyond the scope of this paper, we assume that personal health records describing an adverse reaction to a vaccine are highly sensitive private material. Consequently, they must be protected using techniques guaranteeing the strictest data privacy.

In the FL setting, these data records would be hosted in a participating pharmaceutical company's silo. The pharmaceutical company's silo performs the role of a user in the federation. We view Differential Privacy (DP) as an appropriate tool to enforce

privacy guarantees to individuals' health records. However, more careful analysis of how DP is enforced in FL settings is required. Other technologies such as secure multi-party computation [Yao(1986)] and homomorphic encryption [Gentry(2009)] may be worth considering, but are beyond the scope of this work. Additional security technologies such as end-to-end encryption may be necessary to augument to the DP solution, but is also outside the scope of this work.

We assume a trust model where users do not trust the federation server, and enforce DP *locally* on the parameter updates shipped back to the server. To enforce DP locally, we use the algorithm proposed by Abadi et al. [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, an that injects gaussian noise (calculated using their moments accountant algorithm) in parameter gradients during local training at each user. Noisy gradients lead to noisy parameter updates, which are eventually shipped from the user to the federation server.

Interestingly, since users can possess datasets with different sizes, the computed noise, which is a function of the dataset size, varies considerably from user to user. For instance, the noise introduced for a user with a handful of data points is much higher than the noise introduced by a user with a much larger private dataset. However, FedAvg smoothes out the noisy updates through the parameter aggregation process (averaging, in our case). The resulting model that each user receives is much more robust. Note that our implementation of DP covers the privacy of each narrative, but we assume that there is not enough information in the data to link multiple narratives relating to the same person.

3.2 Mixture of Experts

At its core, the Mixture of Experts proposal is to mix the outputs of a collaboratively-learned general model and a local domain expert model. Participating users have their independent set of labeled training examples that they wish to keep private, drawn from user-specific domain distributions. These users collaborate to build a general model for the task. At the same time, users maintain private, domain-adapted expert models. The final predictor for each user is a weighted average of the outputs from the general and private models. These weights are learned using a MoE architecture [Masoudnia and Ebrahimpour(2014), Nowlan and Hinton(1991)], so the entire model can be trained with gradient descent.

Our overall system architecture follows. M_G is the general model that is trained by the FL framework. (ε, δ)-differential privacy is enforced by clipping gradients and adding noise to the gradients sent back to the federation server. Each of the i users maintains a private model, M_{P_i}, which act as domain experts tuned to the respective users' data distributions. A user uses its private labeled data to retrain its private model M_{P_i} along with M_G. M_{P_i} is completely private to the user and hence does not need DP-inducing noise. The MoE framework at each user combines outputs of the two models, tuning the MoE output to better suit the user's data distribution if needed.

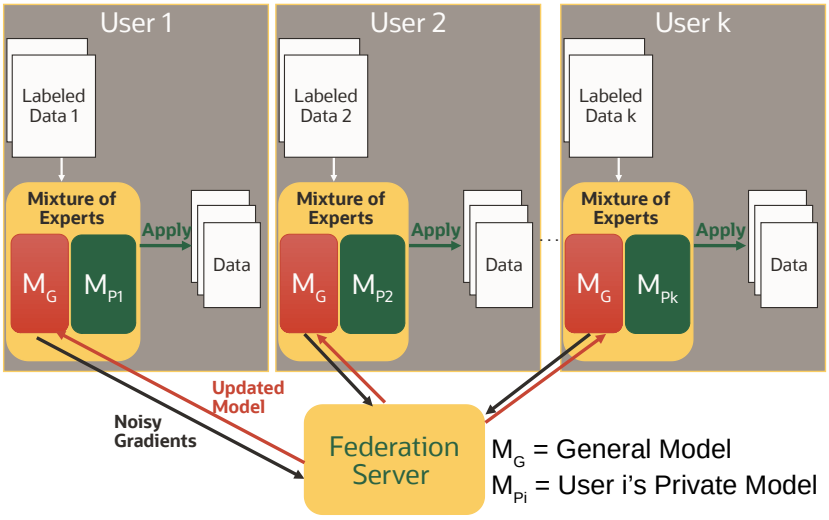

Fig. 3 The Federated Learning setting with Domain Experts, one per participating user. Each user uses its local MoE combination model for both training and inference.

More formally, let M_G be a general model, with parameters Θ_G, so that $\hat{y}_G = M_G(x, \Theta_G)$ is M_G's predicted probability for the positive class, or perhaps a regressed value[1]. M_G is shared between all users, and is trained on all data using FL with differentially private SGD [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang], enabling each user to contribute to training the general model[2].

Similarly, let M_{P_i} be a private model of user i, parameterized by Θ_{P_i}, and $\hat{y}_{P_i} = M_{P_i}(x, \Theta_{P_i})$ be the model's predicted probability. Although M_{P_i} could have a different architecture from M_G, in this work we initialize M_{P_i} as an exact copy of M_G. Neither M_{P_i}, nor gradient information about it, is shared with any other party, so M_{P_i} can be updated exactly, without including privacy-related noise.

These two models are combined using a *gating function*, $\alpha_i(x)$, that can learn which model to trust as a function of the input. In our experiments, we set $\alpha_i(x) = S(w_i^T \cdot x + b_i)$, where $S(x)$ is the sigmoid function, and w_i and b_i are learned weights.

[1] Although we tested only binary classification and regression in our experiments, there are obvious extensions to multiclass problems.

[2] Because M_G is trained using DP SGD protocols, we can guarantee differential privacy of the model M_G. No data besides these differentially-private gradients is ever sent over public channels, so our architecture guarantees differential privacy for anyone without access to the private models M_{P_i}.

Algorithm 1 Minibatch Update for Mixture of Experts at user i (Outline).

inputs
  User i, their N examples $x_1, x_2, \ldots, x_N$ and corresponding labels $y_1, y_2, \ldots, y_N$, along with labeled held out examples H_t to train the gating function, general model M_G and its parameters Θ_G, private model M_{P_i} and its parameters Θ_{P_i}, gating function α_i and its parameters Θ_{α_i}, learning rate η_t, noise scale σ, group size L, gradient norm bound C
Initialize $\mathbf{g}$, $\mathbf{g}_G$, $\mathbf{g}_{P_i}$ to 0.
Take a random sample of data, L_t, with sampling probability L/N
for $x_t \in L_t$ **do**
  Compute prediction $\hat{y}_t$ with equation 3
  Compute loss $l_t = ||y_t - \hat{y}_t||$
  $\mathbf{g}_G += compute\_clipped\_gradient(l_t, \Theta_G, C)$
  $\mathbf{g}_{P_i} += compute\_gradient(l_t, \Theta_{P_i})$
end for
Take a sample of held-out data, H_t
for $x_t \in H_t$ **do**
  Compute prediction $\hat{y}_t$ with equation 3
  Compute loss $l_t = ||y_t - \hat{y}_t||$
  $\mathbf{g}_{\alpha_i} += compute\_gradient(l_t, \Theta_{\alpha_i})$
end for
$\mathbf{g}_G += gaussian\_noise(0, C * \sigma^2)$
$\Theta_{P_i} \leftarrow \Theta_{P_i} - \frac{\eta_t}{L}\mathbf{g}_{P_i}$
$\Theta_{\alpha_i} \leftarrow \Theta_{\alpha_i} - \frac{\eta_t}{L}\mathbf{g}_{\alpha_i}$
Send $\mathbf{g}_G/L$ to the Federation Server.

Although there are many other choices for the gating function, this choice is simple, differentiable, and allows smooth mixing across the boundary between the two models.

Thus the final output $\hat{y}_i$ depends on learned parameters Θ_G, Θ_{P_i}, w_i, and b_i, and all are updated via SGD. The final output that user i uses to label data is

$$\hat{y}_i = \alpha_i(x)M_G(x, \Theta_G) + (1 - \alpha_i(x))M_{P_i}(x, \Theta_{P_i}). \tag{3}$$

We advise, in line with standard MoE practice, that the training of the gating function parameters is separated from the training of the training of the expert models themselves. In all experiments, we split the training data in two and train the parameters of $\alpha_i(x)$ separately from the models M_G and M_{P_i}. We call this held-out data H_t in the pseudocode below.

Algorithm 1 depicts a single batch of MoE training for user i at a high level. During training, each user will perform this procedure many times, while receiving updates to the general model from the federation server. The differentially-private gradient computation $\mathbf{g}_G$ is based on algorithm by Abadi et. al. [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, a
We apply equation 3 to control the gradient aggregate $\mathbf{g}$ that is eventually sent to the federation server.

The private model M_{P_i} and weighting mechanism α_i work together to provide a significant benefit over differentially private FL. First, by allowing individual domain adaptation, they boost accuracy. Second, because they allow noise-free updates, they

prevent the accuracy loss associated with more stringent privacy requirements, which add noise to the general model.

Over time, a user's gating function $\alpha_i(x)$ learns whether to trust the general model or the private model more for a given input, and the private model M_{P_i} needs to perform well on only the subset of points for which the general model fails. The failure of the general model is expected, and corrected, so the mixture of experts can tolerate a poor-quality general model so long as it is reliably correct in at least some region of the input space.

3.2.1 Effects of the Gating Function on Gradients

One advantage of our MoE setup is that the loss is differentiable with respect to all model parameters - the parameters of the private and general models, and the gating function. When $\alpha_i(x) << 1$, the general model makes little contribution to the user's prediction. But this gating function also gates the gradients, because the general model makes little contribution to the loss. Symmetrically, the gradients for the private model vanish as $\alpha_i(x) \rightarrow 1$.

For the private model, gating the gradients is perfectly reasonable. Once the gating function is sure that the general model is making accurate predictions on a particular data point, there is no need to penalize the private model for its performance in that region. The tradeoff in model quality is not symmetric for the shared model, however, because the quality of the shared model affects multiple users. Especially if new users are joining the federation over time, it is still desirable that the general model performs as well as possible while satisfying privacy concerns.

On the other hand, allowing the gating function to suppress updates to the general model may be beneficial for the privacy of user data. Once a users' gating function has learned to trust their private model on a particular data point, there is little incentive for them to share further information about that data point with the federation pool. However, if the overall quality of the general model is not a part of the objective, it only needs to make satisfactory recommendations in a small region of the input space, particularly the regions commonly shared amongst several users. Different regions of reliability correspond to different local optima, and there are many locally-optimal solutions where the general model is an extremely poor model for the overall task, but is still a useful expert for most users on some portion of inputs. We characterize this phenomenon by observing the behavior of the model during training, and evaluation of the shared model on a held-out dataset.

We examine the effects of allowing the gating function to suppress gradients by implementing an alternative algorithm, that still uses an MoE framework. In the modified algorithm, users compute the loss and gradient of the general model on their data points, and send gradients according to the DP SGD procedure, regardless of the weight $\alpha_i(x)$. Then they compute the final prediction using the MoE, and make noise-free updates to their private model and gating function as before. This makes the training of the general model equivalent to traditional FL, while allowing the flexibility of the MoE to improve predictions for each user. This change is

implemented by computing, in addtion to l_t, a loss

$$l_t^G = ||y_t - M_G(x_t, \Theta_G)||,$$

and replacing line 6 of Algorithm 1 with

$$\mathbf{g}_G + = compute\_clipped\_gradient(l_t^G, \Theta_G, C).$$

We refer to this modified version as Algorithm 2.

3.3 Personalization through Fine Tuning

The main allure of FL for a user is the promise of significant prediction accuracy improvements over a locally trained *individual* model. While parameter aggregation through FL can significantly improve accuracy of the global model, introduction of noise to enforce DP can severely compromise that improvement. The degradation can be severe enough to make users reconsider their decision to join the federation, and deter new users from joining the federation. Furthermore, data distributions across users may have significant side effects on the global model's prediction accuracy: If a user's dataset has a significantly different distribution than most of the federation users, the global model may perform worse than a locally trained individual model. If users of a federation have non-IID data, the resulting global model may be ineffective [Li et al.(2020b)Li, Huang, Yang, Wang, and Zhang].

Many researchers have recently proposed different forms of *personalization* approaches to remedy the disparate data distribution problem [Arivazhagan et al.(2019)Arivazhagan, Aggarwal, Singh, and C
Deng, Kamani, and Mahdavi(2020), Jiang et al.(2019)Jiang, Konecný, Rush, and Kannan,
Liang et al.(2020)Liang, Liu, Liu, Salakhutdinov, and Morency, Mansour et al.(2020)Mansour, Mohri, Ro, and Suresh,
Peterson, Kanani, and Marathe(2019), Smith et al.(2017a)Smith, Chiang, Sanjabi, and Talwalkar,
Yu, Bagdasaryan, and Shmatikov(2020)]. Just two of these works [Peterson, Kanani, and Marathe(2019),
Yu, Bagdasaryan, and Shmatikov(2020)], to the best of our knowledge, propose personalization approaches as solutions to model degradation due to DP inducing noise. Among the proposed personalization approaches, we focus on FL with *Fine Tuning* [Yu, Bagdasaryan, and Shmatikov(2020)]: FT-FL for fine tuning on top of plain FL, and FT-DP-FL for fine tuning on top of FL with DP enforcement at the user. In this approach each user continues training, without noise, the local copy of the global differentially private model *after* the FL training process has completed.

The fine tuning based parameter updates are private to each user and are not shared with the federation. As a result, the fine tuned local models may diverge from the global model at varying degrees in order to better fit the users' private data. While endlessly fine tuning the global model can lead to the model converging to a locally trained individual model, care must be taken to ensure that the fine-tuned model does not deteriorate. This can be achieved through standard hyperparameter tuning techniques.

4 Experiments

4.1 Synthetic Dataset

The first dataset is a synthetic regression problem. Two users attempt to fit a linear model of the function $f(x, y) = 5x - 2y + 0.5y^3$. Each has input data drawn from a distinct 2-dimensional Gaussian, and because of these domain differences, they get different exposure to the nonlinear y^3 term. We draw 2500 training examples, 500 validation examples, and 500 test examples for each user, all from that user's 2d Gaussian, then compute $f(x, y)$. The users aim to minimize root mean squared error (RMSE) on the test set. The baseline error is computed with each user fitting a single linear model to their training data. We then compute RMSE for each user if the users collaborate to build a single linear model using FL, and augmenting FL with private domain experts (Algorithm 1). Figure 4 shows the synthetic data, the target function, and the learned gating functions for both users. To see the effects of differential privacy, we test with low noise ($\sigma = 2$) and high noise ($\sigma = 4$), following prior work [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang].

Test errors for the baseline, FL, and domain-adapted FL systems are provided in Table 5. Algorithm 1 provides the best results of any model, and graceful degradation compared to differentially-private FL as the noise increases. FL alone provides a lower error for both users if there are no privacy concerns, but as we increase the noise we apply to the gradient, we observe a dramatic increase in error. The system with domain experts is more expressive than a single linear model - it learns a linear model and gating function for each user on top of the shared linear model - so it is unsurprising that RMSE is lower when no noise is added to the gradients. However, Algorithm 1 does not degrade in performance as much as FL when noise is added to the shared gradient updates. In the worst case, the performance degrades only to the baseline level (where each user has a linear model for its entire dataset).

4.2 Spam Detection Dataset

The second dataset is a real-world domain adaptation dataset for spam detection, which was released as part of the ECML PKDD 2006 Discovery Challenge [Bickel(2006)]. The task is to classify whether an email in a user's inbox is spam, and personalizing the spam filtering for each user. The amount of data available per user is limited, so it is expected that collaboration can increase the quality of the classifier. However, each user has a different inbox, so domain adaptation is required. The dataset was originally designed to test methods of unsupervised domain adaptation, but using the evaluation dataset labels, which are now publicly available, we simulate 15 users collaborating to build a spam classifier in a supervised setting. In this case, we measure classifier accuracy, averaged across all users. We use the dataset from task *b*. For each of our users, we train on 50 labeled examples, leaving 350 examples for testing. The baseline

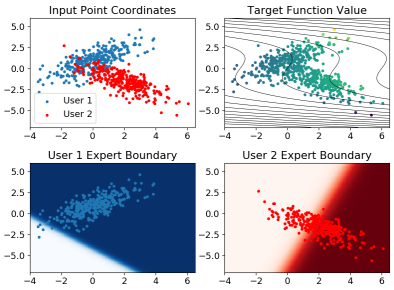

Fig. 4 Visualizing the synthetic data experiment. Axes for all figures represent the (x, y) coordinates of data points. Left to right: (a) (x, y) coordinates of test data points sampled from distinct 2d Gaussians. (b) Target values of nonlinear function $f(x, y)$. (c) Values of the MoE gating function, $\alpha_1(x, y)$ learned by User 1. In the darker region, the private domain expert is preferred, while the general model is preferred in the lighter region. (d) The gating function $\alpha_2(x, y)$ of User 2, which uses the shared model in a different region than User 1.

System	User 1 RMSE	User 2 RMSE
Baseline	15.32	10.95
FL, $\sigma = 0$	12.75	9.67
FL, $\sigma = 2$	13.79	12.68
FL, $\sigma = 4$	12.59	19.49
Alg 1, $\sigma = 0$	12.12	**9.41**
Alg 1, $\sigma = 2$	**12.05**	9.73
Alg 1, $\sigma = 4$	13.78	10.95

Fig. 5 Test error for regression models trained on synthetic data (lower is better). The domain-only baseline system trains a separate model for each user on their data. Varying σ provides different levels of privacy.

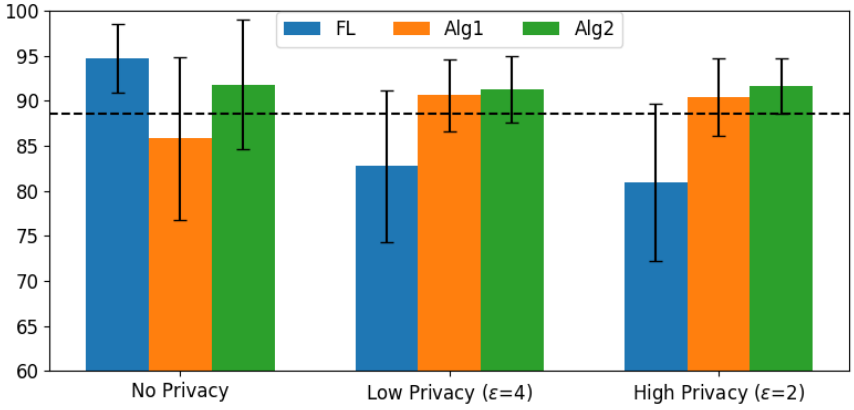

Fig. 6 Classifier accuracy on the spam dataset (higher is better). The dashed horizontal line indicates mean domain-only baseline performance. Error bars show variance across users.

system trains one classifier for each user, using in-domain data only, and we also train a collaborative FL model, and domain-adapted FL models using both Algorithm 1 and Algorithm 2.

In our experiments, we fix $\delta = 10^{-5}$ and compute σ using the moments accountant [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang] to provide guarantees of $\epsilon \in [2, 4]$, simulating relatively strict and relatively permissive differential privacy guarantees. Each system has hyperparameters (learning rate, number of epochs, optimization algorithm) tuned using grid search on an identical validation set; we report average performance on the test set for the winning hyperparameters. As we vary the number of training epochs in our differential privacy paradigms, we also adjust the noise added to make full use of our privacy budget by the end of training. Because the data set is relatively small, our experiments (including hyperparameter optimization) were carried out on an Oracle X6-2 server with 2 CPU sockets.

The results are illustrated in Figure 6. Once again, FL with domain experts provides the best overall accuracy, and maintains its performance as noise is added to provide differential privacy. We see a small increase in accuracy using Algorithm 2, rather than Algorithm 1, across the board. This is especially pronounced in the case where there is no enforcement of privacy, where Algorithm 1 appears to have overfit to a local minima, and test performance is significantly worse. Ordinary FL degrades quickly as we add privacy-protecting noise to the gradient updates, but the MoE domain adaptation technique can clearly protect the accuracy of the per-user predictions, even when noise has degraded the accuracy of the shared FL model.

4.3 MNIST Dataset

MNIST is a widely used image classification dataset that does not contain domain specific partition of the data. Its evaluation however helps us determine if our domain adaptation technique may have wider applicability. To that end, we partition MNIST's training set (with $60,000$ images of hand-written digits) into 100 disjoint partitions, each of which is allotted a unique user. We thus create a federation of 100 users, where each user hosts 600 MNIST images. Each user does a $80/20\%$ training/validation split of its data. We train a linear neural network model using the MNIST training data. MNIST test data is similarly equally split between the 100 users. For differential privacy, we enforce the $\epsilon = 2, 4$ bounds for high, low privacy respectively.

Figure 7 depicts the performance of various models on the MNIST dataset. In the absence of privacy guarantees, the FL model as well as our domain adaptation outperform the baseline domain-only isolated model per user by a significant margin. Their performance however significantly degrades due to DP related noise. Despite the degradation, differentially private FL, as well as the $Alg2$ variation of our algorithm continue to outperform the baseline. The worse performance of $Alg1$ indicates the importance of sharing full gradients to train the general model. Experiments indicate

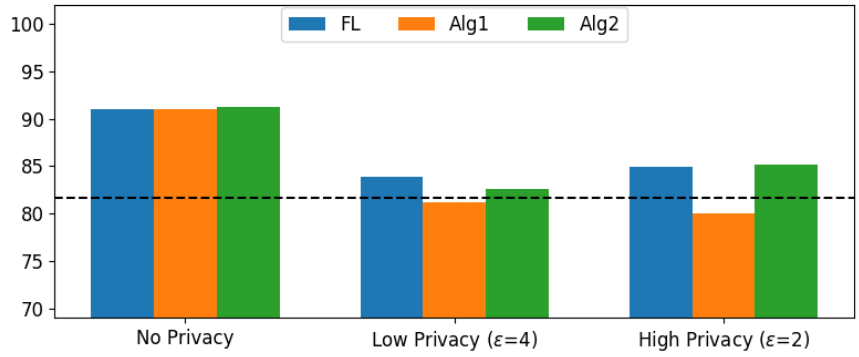

Fig. 7 Classifier accuracy on the MNIST dataset (higher is better). The dashed horizontal line indicates mean domain-only baseline performance.

that our gating function tends to favor the better performing model between the differentially private general FL model and the user-private model.

4.4 Vaccine Adverse Event Reporting System (VAERS) Dataset

Drug and vaccine safety surveillance relies predominantly on spontaneous reporting systems. These systems are comprised of reports of suspected drug/vaccine adverse events (potential side effects) collected from healthcare professionals, consumers, and pharmaceutical companies, and maintained largely by regulatory and health agencies. Among other, these systems are used to detect possible safety problems – called "signals" – that may be related to a vaccination or the consumption of a drug. In the US, the prominent surveillance system for vaccines is the U.S. Centers for Disease Control and Prevention (CDC) and the Food and Drug Administration (FDA) Vaccine Adverse Event Reporting System (VAERS), created in 1990.

The VAERS data (de-identified) is publicly available in structured format. Each VAERS report includes the name of (and additional information about) the administrated vaccine, a list of adverse events related to the vaccine, dates, and limited demographic information about the patient receiving the vaccine (e.g., age, gender). Importantly, the report also includes a textual narrative describing the adverse event. For example,

"*Shortly after patient was vaccinated, she started to feel an itching, tingling feeling in her throat. Fearing that it was an allergic reaction, I called 911. The patient remained alert, talking and breathing normally until paramedics arrived, though she stated that she started to feel additional tingling in her arms and chest.*"

In this example, the following token spans would be annotated as adverse events: “itching”, “tingling feeling in her throat”, “allergic reaction”, “tingling in her arms and chest”.

Most of the data collected in VAERS is currently processed by humans for downstream applications. Adverse event reports, whether they’re forms, emails, articles, or other source documents, do not arrive in structured format, which means they have to be entered manually into safety systems. This manual data entry can take hours and represents a significant cost to the organization. Free-text narratives take the most time, requiring a manual sift through every sentence to find relevant information and then enter it into the correct field. With the rapidly increasing volume of such data this human effort is becoming prohibitive and calls for the increased use of automated methods such as NER. In addition, pharmacovigilance data such as that available in and similar to VAERS originates from private siloed sources, motivating the need for privacy preserving distributed approaches such as FL.

4.4.1 NER based on Recurrent Neural Networks

The recurrent neural network (RNN) architecture we used to perform NER is based on a commonly applied BiLSTM architecture. The architecture consists of three major components: (1) a word representation layer made of word embeddings, (2) two stacked layers of bidirectional long short-term memory (LSTM) cells, and (3) a feedforward layer that performs the final BIO sequence labeling.

Pre-trained word embeddings were used to seed the network’s word embedding layer. These were generated using Word2Vec applied to the sentences comprising the VAERS NER dataset described in section 4. Dropout regularization was implemented between each of the three major network components. The dropout rate was 0.4.

The network was implemented on PyTorch6 and trained using stochastic mini-batch gradient descent with the Adam optimizer for a pre-defined number of iterations. Each iteration processed a batch of 256 randomly selected sentences. The network was trained for a total of 20 epochs, each epoch consisting of number of sentences in the training set / batch size iterations.

4.4.2 Dataset

We used a total of 17,841 narratives submitted to VAERS through the years 2015-2017 to form the NER data set used for this study. The narratives were automatically annotated for adverse event named entities using the list of adverse events supplied with each report. In total the NER data set used for this study comprised of 87,730 sentences and 39,139 annotated adverse event named entities. In our experiments, we split the data randomly into train, validation, tune and test sets in the proportion 60%, 10%, 10%, and 20% respectively. We used the validation set to decide early stopping in the fine tuning algorithm and tuned the rest of parameters on the tune set. We refer to “large manufacturers” as those with more than 1000 VAERS reports in this data and

"small manufacturers" as those with fewer reports to reflect the availability of training data in each user's silo. In the rest of this paper, we use the terms 'manufacturer' and 'user' interchangeably.

4.4.3 Experimental Setup

As the first baseline for our experiments, we train Individual models (*Ind*), i.e. assume that each manufacturer only uses their own training set, and test on their respective test set. This baseline represents the case in which the manufacturer chooses not to participate in the federation at all. *FL* is the federated learning model trained in a collaborative fashion across users using the FedAvg algorithm. This model is then fine tuned for each user using the protocol described in section 3, which yield a set of models, one per manufacturer, that we call *FT*. Next, we introduce DP to the *FL* model, as described in section 3. We use $\epsilon = 2.0$ for this first set of experiments as it is considered a fairly conservative privacy setting in the literature [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang] and calculate the sigma values suitable per user. We call this private federated learning variant *DP-FL*. Finally, we fine tune this private FL model and call it *FT-DP-FL*.

The training parameters for all of these algorithms were tuned using a separate tuning dataset. We use a learning rate of 0.01 and train all the federated models for 20 rounds of FedAvg, with additional 20 epochs for the fine tuning variants at each manufacturer. For evaluation, we compute the precision, recall, and F1 of each token label on a 1-vs-all basis. The values reported are the mean F1 score (henceforth called F1) for the labels at the beginning or inside of an adverse event mention.

We ask the following questions as part of this study. Does *FL* perform better than *Ind* models across users? What happens when differential privacy is introduced? Does personalization help improve accuracy over *FL* and mitigate *DP-FL*'s accuracy loss enough to re-incentivize users to participate in the federation? If fine-tuning based personalization helps mitigate accuracy loss due to DP, how robust is it to varying parameters of DP? Finally, we ask if the federation is stable enough for the uncertainties of real world, such as users dropping out? We also analyze the incentive structure that emerges for users with varying amounts of training data.

4.4.4 Private Federated Learning with Personalization

Figure 8 shows the F1 values for each of the described models on the individual users' test sets. Note that the manufacturers on the x-axis are sorted based on the size of their training sets. As we can see, the FL model consistently outperforms *Ind* models for each of the users, including large manufacturers with a lot of training data. As table 1 shows, the amount of error reduction over the *Ind* model for each user is substantial. Contrary to findings by Yu et. al. [Yu, Bagdasaryan, and Shmatikov(2020)], in our case, personalization based on fine tuning *FT-FL* performs worse than *FL* in most cases. As we add noise related to differential privacy to the federated learning

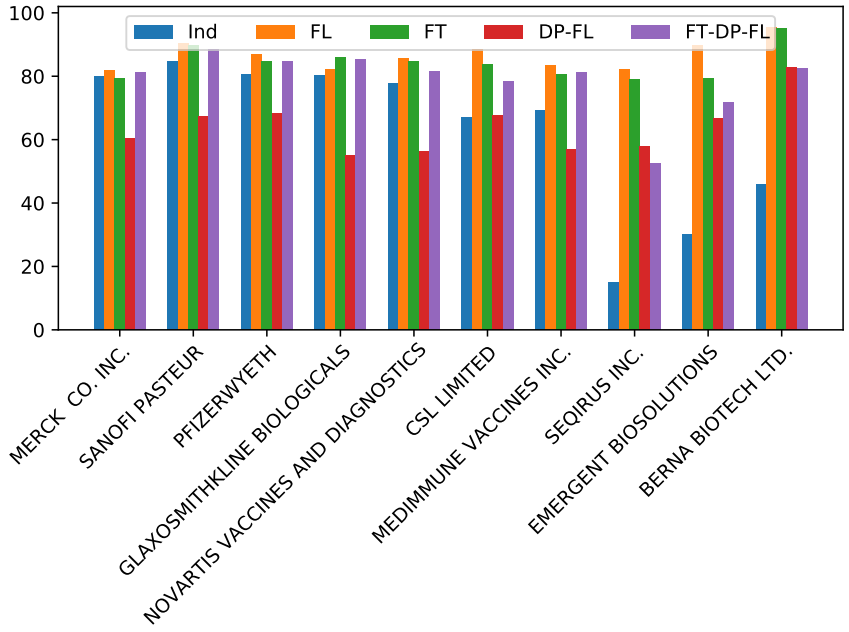

Fig. 8 F1 per manufacturer for different methods for $\epsilon = 2.0$

model, F1 values drop significantly across the board. This makes participation for larger manufacturers in the federation unattractive, since the *DP-FL* model ends up performing worse than their *Ind* models. However, applying fine tuning in this case helps bring it back up to the point, where it is again advantageous for each party to participate in the federation. This shows that personalization based approach can help mitigate the loss of accuracy from introducing differential privacy.

It is interesting to note that for small manufacturers, with an exception of one with very small amount of evaluation data, it is always beneficial to participate in the federation, even for *DP-FL*, with or without personalization. For large manufacturers however, the DP is only attractive in the presence of the mitigation offered by fine-tuning based personalization (*FT-DP-FL*).

5 Related Work

In their work on DP in deep learning, Abadi et. al. [Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar formulated privacy loss of data used to train a deep learning model as a random variable. The moments of this random variable were used to derive tighter bounds for the cumulative privacy loss. The privacy loss was computed in their system using a *moments accountant* module. In our work, we use a subsequent incarnation of the

Vaccine Manufactuer	Num Reports	Individual F1	FL		FT-DP-FL	
			F1	Error Red.	F1	Error Red.
Merck Co. Inc.	7638	80.10	82.00	9.55%	81.20	5.53%
Sanofi Pasteur	3352	84.60	90.40	37.66%	88.40	24.68%
Pfizer-Wyeth	2428	80.50	87.00	33.33%	84.60	21.03%
Glaxo-Smithkline Biologicals	2289	80.20	82.20	10.10%	85.30	25.76%
Novartis Vaccines And Diagnostics	1183	77.80	85.80	36.04%	81.50	16.67%
CSL Limited	465	67.10	88.50	65.05%	78.30	34.04%
Medimmune Vaccines Inc.	265	69.30	83.50	46.25%	81.10	38.44%
Seqirus Inc.	111	15.00	82.10	78.94%	52.60	44.24%
Emergent Biosolutions	58	30.10	89.70	85.26%	71.90	59.80%
Berna Biotech Ltd.	52	45.80	95.40	91.51%	82.50	67.71%

Table 1 F1 and Error Reduction with Federated Learning and Private Federated Learning with Fine Tuning. 'Vaccine Manufacturer' is a field in the public VAERS database that identifies the manufacturer of the vaccine reported in the VAERS form. There is no relationship between this field and the reporter. 'Num VAERS Reports' does not represent the rate of adverse events associated with the manufacturer or its products and cannot be used to estimate such rates. The statistics are based on a sample of reports submitted to VAERS between 2015-2017 whose MedDra coded adverse events appeared in the narrative. Because the statistics are based on a carefully selected sample, the distribution of reports shown may not represent the true distribution of reports associated with different vaccine manufacturers.

moments accountant module [Mironov(2017)] to derive our training data's privacy loss in the general model M_G's training.

Our core idea of maintaining additional models for each domain is analogous to existing domain adaptation approaches [Daumé and Marcu(2006)]. Adjusting federated learning to account for domain differences across users has also been studied in settings without differential privacy concerns [Li et al.(2018)Li, Sahu, Zaheer, Sanjabi, Talwalkar, and Smith, Ji et al.(2018)Ji, Pan, Long, Li, Jiang, and Huang], but these works primarily increase overall model quality by allowing users with extremely unusual gradient updates to have a smaller disruptive effect on the shared FL model. This improves the general model, but does not explicitly allow users with unusual datasets to improve prediction quality on their domain.

Domain adaptation and federated learning have been studied in privacy-preserving and secure settings. One line of work focuses on protecting privacy in a classic domain adaptation setup [Guo et al.(2018)Guo, Yao, Tu, Chen, Dai, and Yang], where a well-tuned model on a source domain is adapted to perform better in a target domain with more limited data. More recently, unsupervised domain adaptation technique for federated learning has been proposed [Peng et al.(2019)Peng, Huang, Zhu, and Saenko], but their setup assumes multiple source domains, and a single target domain, with no joint model trained between users. They also do not take additional privacy measures such as differential privacy into account. The one model per node setting is also explored using multi-task learning [Smith et al.(2017b)Smith, Chiang, Sanjabi, and Talwalkar], but they also do not consider additional privacy in federated learning. Another line of work focuses on secure federated learning [Liu, Chen, and Yang(2018)], but uses additively homomorphic encryption to ensure privacy in a two-party federated learn-

ing context. This is different from ε-differential privacy, and does not maintain a collaborative general model. Each of these systems considers one part of our set-up, but no prior work combines efforts of collaborative learning combined with private domain adaptation.

The PATE architecture [Papernot et al.(2018)Papernot, Song, Mironov, Raghunathan, Talwar, and Erlingsson] is yet another class of distributed ML systems that uses privately trained models of participating parties as sources for consensus-based labeling of data for a new user to help it train its model on its private data. The models trained for individual users act as an ensemble of "teachers" for the new party that is training a new model for itself. The consensus based labeling provides the privacy guarantees for each party. This approach could be used to build the general model, rather than differentially-private FL, but we have not yet tested its effectiveness in conjunction with domain-adaptation techniques.

6 Conclusion

Federated Learning is a promising approach for breaking down organizational and geographical barriers to collaboration on building very effective models to solve this problem. This work demonstrates that adding private, per-user domain adaptation to a collaborative model-building framework can increase accuracy for all users, and is especially beneficial when privacy guarantees begin to diminish the utility of the collaborative general model.

Our implementation of domain adaptation employs a mixture of experts, with each user learning a domain expert model and a private gating mechanism. This domain adaptation framework is another contribution of our work, and allows us to train the entire model with gradient descent. We demonstrate that it works well in practice on both regression and classification tasks. We also apply fine tuning based personalization technique to real world dataset to show similar trends.

In future work, we aim to expand our analysis to include larger datasets and different architectures. We intent do consider other mechanisms for building a collaborative model (e.g., PATE), and alternative domain adaptation techniques (e.g., hypothesis transfer learning). We expect that the general setup of learning one collaborative generalist and a private domain adaptation mechanism will be useful in many settings and for many types of models, but that the best particular architecture could depend on the task. In situations where data privacy is not a concern, for example, the best performance may come from training an FL system and then adding domain adaptation, even with the MoE architecture we propose here, because joint training may overfit the MoE before the general model has had adequate time to learn a high-quality model.

References

Abadi et al.(2016)Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, and Zhang. Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318.

Arivazhagan et al.(2019)Arivazhagan, Aggarwal, Singh, and Choudhary. Arivazhagan, M. G.; Aggarwal, V.; Singh, A. K.; and Choudhary, S. 2019. Federated Learning with Personalization Layers. *CoRR* abs/1912.00818. URL `http://arxiv.org/abs/1912.00818`.

Bagdasaryan et al.(2020)Bagdasaryan, Veit, Hua, Estrin, and Shmatikov. Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How To Backdoor Federated Learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, 2938–2948. PMLR.

Ben-David et al.(2010)Ben-David, Blitzer, Crammer, Kulesza, Pereira, and Vaughan. Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. W. 2010. A Theory of Learning from Different Domains. *Machine Learning* 79(1-2): 151–175. ISSN 0885-6125.

Bickel(2006). Bickel, S. 2006. ECML-PKDD Discovery Challenge 2006 Overview.

Bonawitz et al.(2019)Bonawitz, Eichner, Grieskamp, Huba, Ingerman, Ivanov, Kiddon, Konecný, Mazzocchi, McMahan, Overveldt, Petrou, I
Bonawitz, K.; Eichner, H.; Grieskamp, W.; Huba, D.; Ingerman, A.; Ivanov, V.; Kiddon, C.; Konecný, J.; Mazzocchi, S.; McMahan, H. B.; Overveldt, T. V.; Petrou, D.; Ramage, D.; and Roselander, J. 2019. Towards Federated Learning at Scale: System Design. *CoRR* .

Carlini et al.(2019)Carlini, Liu, Erlingsson, Kos, and Song. Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *28th USENIX Security Symposium*, 267–284.

ccpa(). ccpa. ???? California Consumer Privacy Act (CCPA). *https://oag.ca.gov/privacy/ccpa*.

Chaudhuri, Monteleoni, and Sarwate(2011). Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially Private Empirical Risk Minimization. *The Journal of Machine Learning Research* 12: 1069–1109.

Crammer, Kearns, and Wortman(2008). Crammer, K.; Kearns, M.; and Wortman, J. 2008. Learning from Multiple Sources. *Journal of Machine Learning Research* 9: 1757–1774.

Daumé and Marcu(2006). Daumé, III, H.; and Marcu, D. 2006. Domain Adaptation for Statistical Classifiers. *Journal of Artificial Intelligence Research* 26(1): 101–126. ISSN 1076-9757.

Daumé III(2009). Daumé III, H. 2009. Frustratingly Easy Domain Adaptation. *CoRR* abs/0907.1815. URL `http://arxiv.org/abs/0907.1815`.

Deng, Kamani, and Mahdavi(2020). Deng, Y.; Kamani, M. M.; and Mahdavi, M. 2020. Adaptive Personalized Federated Learning. *CoRR* abs/2003.13461. URL `https://arxiv.org/abs/2003.13461`.

Differential Privacy Team(2017). Differential Privacy Team. 2017. Learning with Privacy at Scale, *https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html*.

Dimitrakakis et al.(2017)Dimitrakakis, Nelson, Zhang, Mitrokotsa, and Rubinstein.
Dimitrakakis, C.; Nelson, B.; Zhang, Z.; Mitrokotsa, A.; and Rubinstein, B. I. P. 2017. Differential Privacy for Bayesian Inference Through Posterior Sampling. *The Journal of Machine Learning Research* 18(1): 343–381.

Dinh, Tran, and Nguyen(2020). Dinh, C. T.; Tran, N. H.; and Nguyen, T. D. 2020. Personalized Federated Learning with Moreau Envelopes. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, virtual*.

Dwork(2006). Dwork, C. 2006. Differential Privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP*, 1–12.

Dwork et al.(2006)Dwork, McSherry, Nissim, and Smith. Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, 265–284.

Dwork and Roth(2014). Dwork, C.; and Roth, A. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends in Theoretical Computer Science* 9(3–4): 211–407.

Fallah, Mokhtari, and Ozdaglar(2020). Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized Federated Learning: A Meta-Learning Approach.

Fredrikson, Jha, and Ristenpart(2015). 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*, 1322–1333.

French, Mackiewicz, and Fisher(2017). French, G.; Mackiewicz, M.; and Fisher, M. 2017. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208* .

gdpr(). gdpr. ???? General Data Protection Regulation (GDPR). *https://gdpr-info.eu/*.

Gentry(2009). Gentry, C. 2009. Fully Homomorphic Encryption Using Ideal Lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, 169–178.

Geyer, Klein, and Nabi(2017). Geyer, R. C.; Klein, T.; and Nabi, M. 2017. Differentially Private Federated Learning: A Client Level Perspective. *CoRR* abs/1712.07557.

Giorgi and Bader(2018). Giorgi, J. M.; and Bader, G. D. 2018. Transfer learning for biomedical named entity recognition with neural networks. *Bioinformatics* 34(23): 4087–4094.

Guo, Shah, and Barzilay(2018). Guo, J.; Shah, D. J.; and Barzilay, R. 2018. Multi-Source Domain Adaptation with Mixture of Experts. *arXiv preprint arXiv:1809.02256* .

Guo et al.(2018)Guo, Yao, Tu, Chen, Dai, and Yang. Guo, X.; Yao, Q.; Tu, W.; Chen, Y.; Dai, W.; and Yang, Q. 2018. Privacy-preserving Transfer Learning for Knowledge Sharing. *arXiv preprint arXiv:1811.09491* .

Gurulingappa et al.(2012)Gurulingappa, Rajput, Roberts, Fluck, Hofmann-Apitius, and Toldo. Gurulingappa, H.; Rajput, A. M.; Roberts, A.; Fluck, J.; Hofmann-Apitius, M.; and Toldo, L. 2012. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *Journal of Biomedical Informatics* 45(5): 885–892.

Haerian et al.(2012)Haerian, Varn, Vaidya, Ena, Chase, and Friedman. Haerian, K.; Varn, D.; Vaidya, S.; Ena, L.; Chase, H.; and Friedman, C. 2012. Detection of pharmacovigilance-related adverse events using electronic health records and automated methods. *Clinical Pharmacology and Therapeutics* 92(2): 228–234.

Harpaz et al.(2014)Harpaz, Callahan, Tamang, Low, Odgers, Finlayson, Jung, LePendu, and Shah. Harpaz, R.; Callahan, A.; Tamang, S.; Low, Y.; Odgers, D.; Finlayson, S.; Jung, K.; LePendu, P.; and Shah, N. 2014. Text Mining for Adverse Drug Events: the Promise, Challenges, and State of the Art. *Drug safety : an international journal of medical toxicology and drug experience* .

Hitaj, Ateniese, and Perez-Cruz(2017). Hitaj, B.; Ateniese, G.; and Perez-Cruz, F. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 603–618.

Hsieh et al.(2019)Hsieh, Phanishayee, Mutlu, and Gibbons. Hsieh, K.; Phanishayee, A.; Mutlu, O.; and Gibbons, P. B. 2019. The Non-IID Data Quagmire of Decentralized Machine Learning. *CoRR* abs/1910.00189. URL `http://arxiv.org/abs/1910.00189`.

imi(). imi. ???? Innovatice Medices Initiative: Europe's Partnership for Health. *https://www.imi.europa.eu*.

Ji et al.(2018)Ji, Pan, Long, Li, Jiang, and Huang. Ji, S.; Pan, S.; Long, G.; Li, X.; Jiang, J.; and Huang, Z. 2018. Learning Private Neural Language Modeling with Attentive Aggregation. *arXiv preprint arXiv:1812.07108* .

Jiang et al.(2019)Jiang, Konecný, Rush, and Kannan. Jiang, Y.; Konecný, J.; Rush, K.; and Kannan, S. 2019. Improving Federated Learning Personalization via Model Agnostic Meta Learning. *CoRR* abs/1909.12488. URL `http://arxiv.org/abs/1909.12488`.

Kairouz et al.(2019)Kairouz, McMahan, Avent, Bellet, Bennis, Bhagoji, Bonawitz, Charles, Cormode, Cummings, D'Oliveira, Rouayheb, Ev
Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.;

Konecný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2019. Advances and Open Problems in Federated Learning. *CoRR* abs/1912.04977. URL http://arxiv.org/abs/1912.04977.

Kasiviswanathan et al.(2008)Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith. Kasiviswanathan, S. P.; Lee, H. K.; Nissim, K.; Raskhodnikova, S.; and Smith, A. D. 2008. What Can We Learn Privately? *CoRR* abs/0803.0924. URL http://arxiv.org/abs/0803.0924.

Konecný, McMahan, and Ramage(2015). Konecný, J.; McMahan, B.; and Ramage, D. 2015. Federated Optimization: Distributed Optimization Beyond the Datacenter. *CoRR* abs/1511.03575. URL http://arxiv.org/abs/1511.03575.

Konecný et al.(2016)Konecný, McMahan, Ramage, and Richtárik. Konecný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *CoRR* abs/1610.02527.

Korkontzelos et al.(2016)Korkontzelos, Nikfarjam, Shardlow, Sarker, Ananiadou, and Gonzalez. Korkontzelos, I.; Nikfarjam, A.; Shardlow, M.; Sarker, A.; Ananiadou, S.; and Gonzalez, G. H. 2016. Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. *Journal of Biomedical Informatics* 62: 148–158. ISSN 1532-0464. doi:https://doi.org/10.1016/j.jbi.2016.06.007. URL https://www.sciencedirect.com/science/article/pii/S1532046416300508.

Korolova(2010). Korolova, A. 2010. Privacy Violations Using Microtargeted Ads: A Case Study. In *2010 IEEE International Conference on Data Mining Workshops*, 474–482.

Kouw and Loog(2018). Kouw, W. M.; and Loog, M. 2018. An introduction to domain adaptation and transfer learning. *CoRR* abs/1812.11806. URL http://arxiv.org/abs/1812.11806.

Leaman et al.(2010)Leaman, Wojtulewicz, Sullivan, Skariah, Yang, and Gonzalez. Leaman, R.; Wojtulewicz, L.; Sullivan, R.; Skariah, A.; Yang, J.; and Gonzalez, G. 2010. Towards Internet-Age Pharmacovigilance: Extracting Adverse Drug Reactions from User Posts in Health-Related Social Networks. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2010, Uppsala, Sweden, July 15, 2010*, 117–125. Association for Computational Linguistics.

LePendu et al.(2013)LePendu, Iyer, Bauer-Mehren, Harpaz, Mortensen, Podchiyska, Ferris, and Shah. LePendu, P.; Iyer, S.; Bauer-Mehren, A.; Harpaz, R.; Mortensen, J.; Podchiyska, T.; Ferris, T.; and Shah, N. 2013. Pharmacovigilance Using Clinical Notes. *Clinical Pharmacology and Therapeutics* 93: 547–555.

Li et al.(2018)Li, Sahu, Zaheer, Sanjabi, Talwalkar, and Smith. Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* .

Li et al.(2020a)Li, Gu, Dvornek, Staib, Ventola, and Duncan. Li, X.; Gu, Y.; Dvornek, N.; Staib, L. H.; Ventola, P.; and Duncan, J. S. 2020a. Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. *Medical Image Analysis* 65: 101765.

Li et al.(2020b)Li, Huang, Yang, Wang, and Zhang. Li, X.; Huang, K.; Yang, W.; Wang, S.; and Zhang, Z. 2020b. On the Convergence of FedAvg on Non-IID Data. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Liang et al.(2020)Liang, Liu, Liu, Salakhutdinov, and Morency. Liang, P. P.; Liu, T.; Liu, Z.; Salakhutdinov, R.; and Morency, L. 2020. Think Locally, Act Globally: Federated Learning with Local and Global Representations. *CoRR* abs/2001.01523. URL http://arxiv.org/abs/2001.01523.

Liu, Chen, and Yang(2018). Liu, Y.; Chen, T.; and Yang, Q. 2018. Secure Federated Transfer Learning. *arXiv preprint arXiv:1812.03337* .

London(2020). London, B. 2020. PAC Identifiability in Federated Personalization. In *NeurIPS 2020 Workshop on Scalability, Privacy, and Security in Federated Learning*.

Mansour et al.(2020)Mansour, Mohri, Ro, and Suresh. Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three Approaches for Personalization with Applications to Federated Learning. *CoRR* abs/2002.10619. URL https://arxiv.org/abs/2002.10619.

Masoudnia and Ebrahimpour(2014). Masoudnia, S.; and Ebrahimpour, R. 2014. Mixture of Experts: A Literature Survey. *Artificial Intelligence Review* 42(2): 275–293. ISSN 0269-2821.

McMahan et al.(2016)McMahan, Moore, Ramage, and y Arcas. McMahan, H. B.; Moore, E.; Ramage, D.; and y Arcas, B. A. 2016. Federated Learning of Deep Networks using Model Averaging. *CoRR* abs/1602.05629.

McMahan et al.(2017)McMahan, Ramage, Talwar, and Zhang. McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2017. Learning Differentially Private Language Models Without Losing Accuracy. *CoRR* abs/1710.06963. URL http://arxiv.org/abs/1710.06963.

Melis et al.(2018)Melis, Song, Cristofaro, and Shmatikov. Melis, L.; Song, C.; Cristofaro, E. D.; and Shmatikov, V. 2018. Inference Attacks Against Collaborative Learning. *CoRR* abs/1805.04049. URL http://arxiv.org/abs/1805.04049.

mellody(). mellody. ???? New Research Consortium Seeks to Accelerate Drug Discovery Using Machine Learning to Unlock Maximum Potential of Pharma Industry Data *https://www.-janssen.com/emea/new-research-consortium- seeks-accelerate-drug-discovery-using-machine-learning-unlock-maximum*.

Mironov(2017). Mironov, I. 2017. Renyi Differential Privacy. *CoRR* abs/1702.07476. URL http://arxiv.org/abs/1702.07476.

Nasr, Shokri, and Houmansadr(2019). Nasr, M.; Shokri, R.; and Houmansadr, A. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019*, 739–753. IEEE. URL https://doi.org/10.1109/SP.2019.00065.

Nowlan and Hinton(1991). Nowlan, S. J.; and Hinton, G. E. 1991. Evaluation of adaptive mixtures of competing experts. In *Advances in neural information processing systems*, 774–780.

Pan and Yang(2010). Pan, S. J.; and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10): 1345–1359. ISSN 1041-4347.

Papernot et al.(2018)Papernot, Song, Mironov, Raghunathan, Talwar, and Erlingsson. Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Erlingsson, Ú. 2018. Scalable Private Learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Peng et al.(2019)Peng, Huang, Zhu, and Saenko. Peng, X.; Huang, Z.; Zhu, Y.; and Saenko, K. 2019. Federated Adversarial Domain Adaptation.

Peterson, Kanani, and Marathe(2019). Peterson, D. W.; Kanani, P.; and Marathe, V. J. 2019. Private Federated Learning with Domain Adaptation. *CoRR* abs/1912.06733. URL http://arxiv.org/abs/1912.06733.

Roberts, Demner-Fushman, and Tonning(2017). Roberts, K.; Demner-Fushman, D.; and Tonning, J. M. 2017. Overview of the TAC 2017 Adverse Reaction Extraction from Drug Labels Track. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*. NIST.

Samdani and Yih(2011). Samdani, R. Y.; and Yih, W.-t. 2011. Domain adaptation with ensemble of feature groups. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Shokri et al.(2017)Shokri, Stronati, Song, and Shmatikov. 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)*, 3–18.

Smith et al.(2017a)Smith, Chiang, Sanjabi, and Talwalkar. Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. 2017a. Federated Multi-Task Learning.

Smith et al.(2017b)Smith, Chiang, Sanjabi, and Talwalkar. Smith, V.; Chiang, C.-K.; Sanjabi, M.; and Talwalkar, A. 2017b. Federated Multi-Task Learning.

Sun and Shi(2013). Sun, S.-L.; and Shi, H.-L. 2013. Bayesian multi-source domain adaptation. In *2013 International Conference on Machine Learning and Cybernetics*, volume 1, 24–28. IEEE.

Tramèr et al.(2016)Tramèr, Zhang, Juels, Reiter, and Ristenpart. Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing Machine Learning Models via Prediction APIs. In *Proceedings of the 25th USENIX Conference on Security Symposium*, 601–618.

Tu and Sun(2012). Tu, W.; and Sun, S. 2012. Dynamical ensemble learning with model-friendly classifiers for domain adaptation. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 1181–1184. IEEE.

Winnenburg et al.(2015)Winnenburg, Sorbello, Ripple, Harpaz, Tonning, Szarfman, Francis, and Bodenreider. Winnenburg, R.; Sorbello, A.; Ripple, A.; Harpaz, R.; Tonning, J.; Szarfman, A.; Francis, H.; and Bodenreider, O. 2015. Leveraging MEDLINE indexing for pharmacovigilance - Inherent limitations and mitigation strategies. *Journal of Biomedial Informatics* .

Yao(1986). Yao, A. C. 1986. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science*, 162–167.

Yu, Bagdasaryan, and Shmatikov(2020). Yu, T.; Bagdasaryan, E.; and Shmatikov, V. 2020. Salvaging Federated Learning by Local Adaptation. *CoRR* abs/2002.04758. URL `https://arxiv.org/abs/2002.04758`.

7 Appendix

7.1 Effect of the Gating Function on Gradients

7.1.1 Empirical Gradient Sizes During Training

It is worth getting an estimate of how much the gating function actually affects the gradients passed during training. We measure the average L2-norm of the gradients across each batch as we train the spam detection model for 15 users. We consider both traditional FL, where the gradient sizes are unconstrained, and the DP-SGD procedure, where each example has its gradient clipped to a maximum L2-norm (in our case, the norm of the gradient for each parameter matrix is independently clipped to a magnitude of 1). In both cases, we observe a marked reduction in the magnitude of gradients sent over the course of training, if the gating function is allowed to suppress gradients that would otherwise be sent to the general model.

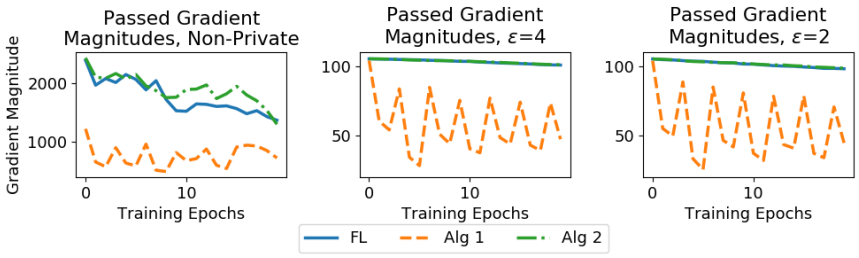

Fig. 9 Average magnitude of gradients passed to the federation server as training progresses. In Algorithm 1, the gating function suppresses the gradients sent to the general model.

It is clear from Figure 9 that allowing $\alpha_i(x)$ to gate the gradients reduces the total magnitude of gradient updates, whether we clip gradients to a bounded magnitude or not, and that Algorithm 2 passes essentially the same magnitude of gradients to the federation server as FL without the mixture of experts. The averages shown here are across multiple restarts and all users.

7.1.2 Stability of the General Model's Predictions

We also wish to test how often the general model M_G gets caught in local optima if the gradients sent to the general model are gated. One way to measure this is to restart the model with another initialization, and evaluate whether it tends to make the same predictions on a fixed dataset. The spam detection dataset includes a small dataset of 100 labeled emails not associated to any particular user. We use this held-out data to analyze how much the shared model varies across runs.

We ran the model with ten distinct random initializations, and compared the pairwise differences in predictions on these held-out emails. We also ran the model with a single fixed initialization, but perturbed the training by holding out a single user at each run. Both types of perturbations are extremely plausible in a real-world FL scenario - we have no guarantees that we're in an optimal random initialization, or that the user base stays fixed.

Model	**Privacy Budget**	$P(\hat{y}_1 \neq \hat{y}_2)$	**Accuracy**
FL	∞	0.38 ± 0.13	0.65 ± 0.11
FL	4	0.38 ± 0.13	0.56 ± 0.06
FL	2	0.43 ± 0.12	0.55 ± 0.07
Alg 1	∞	0.50 ± 0.22	0.53 ± 0.06
Alg 1	4	0.47 ± 0.12	0.51 ± 0.08
Alg 1	2	0.51 ± 0.14	0.50 ± 0.09
Alg 2	∞	0.35 ± 0.12	0.65 ± 0.06
Alg 2	4	0.35 ± 0.08	0.53 ± 0.06
Alg 2	2	0.43 ± 0.10	0.52 ± 0.06

Table 2 Stability of predictions of the shared model across random re-initializations. The shared model makes dramatically different predictions on the same held-out data, an effect which is stronger in Algorithm 1 than FL or Algorithm 2.

Table 2 and Table 3 show that the predictions of the shared model are much more similar across runs if full gradients are passed, compared to allowing the gating function to suppress gradients for "private" data points. The general model is also more accurate using Algorithm 2 compared to Algorithm 1. The instability and inaccuracy of the general model makes only a slight impact on per-user accuracy, though, as seen in Table 6, suggesting that the users are typically able to find satisfactory work-arounds for any given general model, and that Algorithm 1 tends to stop in local optima that differ greatly based on initialization.

Model	Privacy Budget	$P(\hat{y}_1 \neq \hat{y}_2)$	Accuracy
FL	∞	0.31 ± 0.10	0.64 ± 0.09
FL	4	0.37 ± 0.10	0.54 ± 0.07
FL	2	0.44 ± 0.09	0.52 ± 0.07
Alg 1	∞	0.42 ± 0.15	0.59 ± 0.07
Alg 1	4	0.46 ± 0.12	0.48 ± 0.07
Alg 1	2	0.49 ± 0.12	0.47 ± 0.09
Alg 2	∞	0.35 ± 0.11	0.64 ± 0.09
Alg 2	4	0.39 ± 0.11	0.52 ± 0.06
Alg 2	2	0.47 ± 0.10	0.50 ± 0.06

Table 3 Stability of predictions of the shared model with a fixed initialization, as individual users are dropped from the training pool. Again, FL and Alg 2 provide significantly more stable predictions than Alg 1. Overall, variance in predictions due to dropping a user is slightly less than variance due to initialization differences.

Overall, Algorithm 2 provides much more stability, a boost to accuracy on the general model, and a slight improvement in per-user accuracy. These properties are likely to make it more useful on larger datasets.

7.2 Stability of Federation against Users Leaving

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0.0	0.9	1.8	0.4	1.0	2.1	1.8	0.4	1.0	0.0
M2	-0.3	0.0	0.4	0.5	1.4	1.6	1.6	-0.4	3.2	-1.5
M3	-0.1	0.5	0.0	0.1	0.1	0.9	1.4	1.9	1.0	-1.5
M4	-0.6	0.8	0.2	0.0	2.6	-0.2	3.5	1.3	1.0	0.0
M5	-0.5	-0.1	-0.1	2.9	0.0	0.6	0.6	-1.9	1.0	0.0
M6	-0.8	0.0	0.2	-0.5	-0.4	0.0	1.6	-1.1	2.1	0.0
M7	-0.5	0.5	-0.3	-0.5	0.1	0.7	0.0	0.4	1.0	-1.5
M8	-0.7	0.3	0.3	-0.1	-0.5	0.0	-0.5	0.0	0.8	0.0
M9	-0.4	0.1	0.2	0.0	0.4	0.1	0.9	0.9	0.0	4.5
M10	-1.0	0.0	-0.2	-0.2	-0.2	0.3	-1.3	-1.1	0.0	0.0

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
M1	0.0	0.1	0.4	1.9	-2.4	1.4	2.9	-8.3	0.3	15.8
M2	-0.1	0.0	0.6	1.6	-1.5	-1.6	0.5	-2.5	1.4	22.5
M3	0.5	0.5	0.0	2.1	-1.7	0.2	-1.3	-1.2	-1.2	3.7
M4	-0.3	-0.3	0.2	0.0	-0.1	-4.3	0.7	-1.3	-0.4	18.7
M5	-0.1	0.0	-0.3	1.0	0.0	-0.3	-0.3	-1.9	-0.8	0.5
M6	-0.2	-0.5	0.3	1.6	-1.9	0.0	-1.5	-0.3	-0.5	4.2
M7	-0.5	0.1	0.3	2.2	-1.2	-2.8	0.0	-0.5	0.9	28.9
M8	0.5	-0.5	0.8	0.6	0.0	-4.0	-0.9	0.0	5.2	15.8
M9	-0.5	-0.5	0.3	1.0	-2.5	-3.3	-3.5	-2.4	0.0	4.1
M10	-0.1	-0.2	1.0	0.9	-1.8	-3.2	-0.1	-1.4	2.2	0.0

Table 4 Stability of Private FL with Fine Tuning performance when a single user leaves. M1-M10 are manufacturers sorted in descending order by size. Each row represents a manufacturer that is leaving the federation. Each Column represents the difference between F1 values under full federation and this reduced federation for that manufacturer. The table on the left represents FL and the table on the right represents FT-DP-FL

Building a federation across organizations can be challenging in the real world due to a variety of factors. For instance, users may discontinue their participation in the federation. We simulate this scenario and study the effect of one of the manufacturers leaving the federation. As we can see from Table 4, both federated learning and private federated learning with fine tuning are fairly stable against such a change, with the exception of a few manufacturers with very small amount of training and

test data. In other words, no single manufacturer has disproportionally large impact on the overall accuracy gains from participating in the federation.

7.3 Federation of Small Manufacturers

Another scenario that we simulate is the one where only participants with small amount of training data agree to collaborate. In this case, we do not have the advantage of the large amount of training data from any of the larger manufacturers. To better understand if such a federation is still advantageous, we compare the F1 values for small manufacturers in two different scenarios: one, in which they are a part of a large federation with all manufacturers, and second, in which they are a part of a federation with only the small manufacturers.

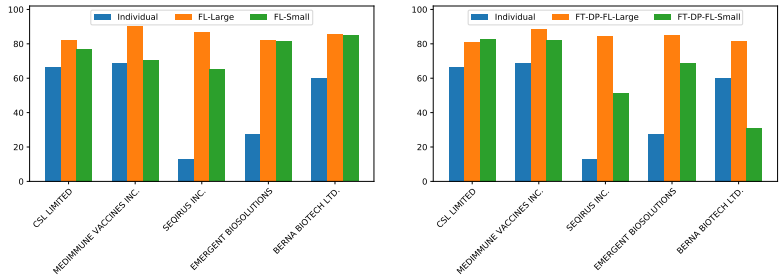

Fig. 10 F1 for small manufacturers when they are a part of a larger federation vs. a federation of only small manufacturers. The graph on left is for FL and the one on right is for FT-DP-FL

Figure 10 shows these comparisons for FL and FT-DP-FL respectively. As is clear from the bar chart, even in the case of a federation with just the small manufacturers, most of the manufacturers benefit significantly from participating. In fact, the performance of all manufacturers in the small federation closely tracks their performance in the large federation, with one exception.

7.4 Robustness to Differential Privacy Noise

Next, we study the effectiveness of personalization in recovering from the accuracy loss resulting from differential privacy noise. We vary the parameter ϵ and measure F1 averaged across users for two of the algorithm variants: differentially private federated learning (DP-FL) and the fine tuned differentially private federated learning (FT-DP-FL). As we can see from Figure 11, average F1 for DP-FL deteriorates significantly for values of ϵ less than 2. However, even in these cases, the personalized version, FT-DP-FL manages to retain its performance. We believe this is an important

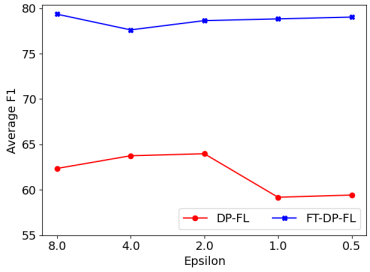

Fig. 11 Average F1 across users for the two differentially private FL variants.

finding that provides significant latitude to differentially private FL frameworks to further tighten the privacy budget of ϵ without compromising utility.

8 PAC Non-Identifiability

Inspired by Probably Approximately Correct (PAC) learning, London recently introduced *PAC Identifiability* [London(2020)], a new privacy condition relevant to personalization in FL [Dinh, Tran, and Nguyen(2020), Fallah, Mokhtari, and Ozdaglar(2020), Liang et al.(2020)Liang, Liu, Liu, Salakhutdinov, and Morency, Mansour et al.(2020)Mansour, Mohri, Ro, and Suresh, Peterson, Kanani, and Marathe(2019), Yu, Bagdasaryan, and Shmatikov(2020)]. Informally, in a personalized FL setting, PAC identifiability determines whether the private model used by a federation's user can be leaked out to an adversarial federation server. Learning a user's private model can fundamentally compromise the user's privacy. London presents the case study of recommender systems, where the federation server may be able to determine ratings choices made by a targeted user. It is critical for user privacy to determine if a given personalization approach's user-local (private) model is PAC identifiable. To that end we now prove that FT-FL is not PAC identifiable.

Let G be the global model containing parameters $p_1, p_2, ..., p_n$. Let L_u be the local (private) model for user u, and D_u denote the user's private data used to train G and L_u. We can w.l.o.g. represent personalization in FL at user u as follows:

$$\Delta p_u = L_u(D_u) \oplus G(D_u) \tag{4}$$

where $\oplus$ is the personalization specific operator (algorithm) that combines the local and global models' outputs to yield Δp_u, the update to G's parameters that is shipped back to the federation server.

We use London's definition of PAC identifiability in his restricted context of binary classification for a recommender system in our proof. However, our proof can be easily generalized to a richer definition of PAC identifiability.

Definition 1 A user u, using a given protocol (which may be stochastic), is *PAC Identifiable* if, for any $\epsilon \in (0, 1)$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$ over $T = poly(\epsilon^{-1}, \delta^{-1})$ interactions with the server, the server can output an estimate $\hat{L}_u$, of the local model (after interaction), L_u, such that

$$\frac{1}{I}\sum_{i \in \mathcal{I}} \mathbb{1}\{\hat{L}_u(D_u) \oplus G(D_u) \neq L_u(D_u) \oplus G(D_u)\} \leq \epsilon.$$

where, $\mathcal{I}$ is the set of I items from the catalog in the recommender system. Informally, PAC identifiability puts an upper bound ϵ on the number of disagreements between the user's local model L_u, and its estimate $\hat{L}_u$ predicted by an adversarial federation server. In such cases, we say that models L_u and $\hat{L}_u$ are *similar*.

While London [London(2020)] describes a simple PAC identifiability mechanism (protocol), more sophisticated mechanisms will be proposed by researchers in the future. Our proof of FT-FL's PAC *non-identifiability* is agnostic to such mechanisms.

Formally, let A_G be a mechanism employed by the adversarial federation server such that

$$A_G(\Delta p_u) \triangleq \hat{L}_u \tag{5}$$

where $\hat{L}_u$ is the estimate of L_u. We say that A_G is the *PAC identifiability mechanism* for L_u.

Clearly, in the process of deriving $\hat{L}_u$, $A_G(\Delta p_u)$ eliminates $G(D_u)$ or its effects from Equation 8. Let us call that operation G^-. Therefore,

$$G^-(\Delta p_u) = L_u(D_u) + \gamma \tag{6}$$

where γ is the noise introduced by G^- in the process of eliminating the effects of $G(D_u)$ on Δp_u. Thus,

$$A_G(\Delta p_u) = U^-(G^-(\Delta p_u)) \tag{7}$$

where U^- maps $L_u(D_u) + \gamma$ to $\hat{L}_u$. γ must be negligible enough to allow the PAC identifiability condition (2) to be satisfied.

Lemma 1 *In any setting where a model is trained by FL, and users fine-tune the model within their silo after training is complete, the adversarial federation server's PAC identifiability mechanism, A_G, yields a model that is similar to the* Null *model (the model with all its parameters set to the value* 0*):*

$$A_G(\Delta p_u) = O$$

Proof As per Equation 8

$$\Delta p_u = L_u(D_u) \oplus G(D_u)$$

In case of FT-FL, $L_u(D_u)$ is completely missing from the above composition that yields the parameter update Δp_u. In fact, $L_u(D_u)$ is computed *after* the entire FL training process completes. Recall however, that $L_u(D_u)$ is used by user u privately to make its post-training local predictions. In effect,

$$\Delta p_u = \bar{0} \oplus G(D_u)$$

In fact, $\Delta p_u = G(D_u)$. As a result, Equation 10 evaluates to

$$G^-(\Delta p_u) = \bar{0} + \gamma$$

and as U^- maps $\bar{0} + \gamma$ to $\hat{L}_u$, the latter is similar to the Null model O.

The following corollaries follow

Corollary 1 *FT-FL is not PAC identifiable.*

Corollary 2 *FT-DP-FL is not PAC identifiable.*

Since fine-tuning of FT-FL (and FT-DP-FL) follows conventional training methodologies, the convergence proof of the fine tuning component of FT-FL (and FT-DP-FL) is identical to standard convergence proofs for stochastic gradient descent and similar optimization algorithms.