Empirical Studies of Institutional Federated Learning For Natural Language Processing

Xinghua Zhu, Jianzong Wang ✉ †**, Zhenhou Hong,** and **Jing Xiao**
Ping An Technology (Shenzhen) Co., Ltd.
Shenzhen, P.R.China
{zhuxinghua889, wangjianzong347, hongzhenhou168,
xiaojing661}@pingan.com.cn

Abstract

Federated learning has sparkled new interests in the deep learning society to make use of isolated data sources from independent institutes. With the development of novel training tools, we have successfully deployed federated natural language processing networks on GPU-enabled server clusters. This paper demonstrates federated training of a popular NLP model, TextCNN, with applications in sentence intent classification. Furthermore, differential privacy is introduced to protect participants in the training process, in a manageable manner. Distinguished from previous client-level privacy protection schemes, the proposed differentially private federated learning procedure is defined in the dataset sample level, inherent with the applications among institutions instead of individual users. Optimal settings of hyper-parameters for the federated TextCNN model are studied through comprehensive experiments. We also evaluated the performance of federated TextCNN model under imbalanced data load configuration.

Experiments show that, the sampling ratio has a large impact on the performance of the FL models, causing up to 38.4% decrease in the test accuracy, while they are robust to different noise multiplier levels, with less than 3% variance in the test accuracy. It is also found that the FL models are sensitive to data load balancedness among client datasets. When the data load is imbalanced, model performance dropped by up to 10%.

1 Introduction

Federated learning is a promising ideology to unite isolated datasets for machine learning problems (Konečnỳ et al., 2016; McMahan et al., 2016; Zhu et al., 2019). In the federated learning framework, no raw data are exchanged among participating entities. Instead, parameter gradients and aggregated updates are communicated between servers during collective optimization. Therefore, without leaking private information, institutes can cooperate with each other by contributing their data collection in the training of a unified model. Such a feature is especially desirable when handling sensitive data that involve e.g. personal preference, financial transactions, medical records, etc. An example of successful deployment of federated learning is the smart input prediction in Google Input (Hard et al., 2018). In addition, more business-to-client model training applications are drawing intensive attention of the public (Lim et al., 2020; Yang et al., 2020; Kong et al., 2020). Apart from this business-to-client cooperation case, more interesting applications can be found among institutions. Potential areas include medical image analysis (Sheller et al., 2018), smart retail (Yang et al., 2019b), fraud detection, etc.

Despite its promising designs, federated learning met quite some difficulties migrating to deeper neural networks, as well as to broader cooperative areas. These difficulties are largely due to

1. the limited training speed offered by a secured federated learning platform; and
2. lack of quantifiable evaluation of the privacy and performance of the federated models.

Before these issues can be settled, institutions would prone to keep their data private rather than contributing to a collaborative neural model.

In terms of system security, a federated learning algorithm needs to take care of two kinds of adversaries. Firstly, the communication between participating servers must be protected from third-party interception or modification. Secondly, local datasets must be protected from probing or reverse engineering by other participants. The communication encryption / decryption, such as AES, 3DESE,

† Cooresponding author: Jianzong Wang (jzwang@188.com)

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 625–634
November 16 - 20, 2020.

RSA etc., is often lossless, therefore does not affect the performance of the trained model itself. On the other hand, the latter security concern about adversaries from within the participants, has to be addressed differently. To protect the anonymity of training data, the models often need to be revised according to the anonymity protection mechanism.

Since the anonymity protection scheme is involved in the training procedure, it also causes changes in the expected performance of the overall system. In previous works, the anonymity protection scheme and its influence on the system performance is analyzed case-by-case. For example, Yang et al. proposed a local clustering to ensure the k-anonymity of the XGBoost model, and studied the relationship between the number of clusters and prediction accuracy via experiments (Yang et al., 2019a). Unfortunately, their results do not generalize to other models or other anonymity metrics. The restricted applicability of such analyses have limited the development of institutional cooperation on federated learning frameworks.

In this paper, we adopt the (ϵ, δ)-differential privacy defined by Dwork (Dwork, 2011) as the universal privacy metric. We extend the model privacy derived in (Abadi et al., 2016) to the federated training model. By utilizing recent developments of federated learning framework, we implemented the federated training of the TextCNN model (Kim, 2014). To our knowledge, this is the first reported implementation of NLP models on federated learning frameworks.

Contributions of this paper include:

1. Adapt the differentially private deep learning algorithm to institutional federated learning framework. Implement differentially private federated TextCNN model for text intent classification.

2. Analyse the performance of federated TextCNN with various differential privacy settings. We show that the differential privacy itself does not negatively affect the performance of the trained models.

3. In an institutional cooperation mode, analyse the performance of federated TextCNN with a wide range of data distribution configurations. It is shown that the accuracies of the trained models are sensitive to the number of data splits, as well as the balancedness of the data distribution.

2 Related Work

2.1 Federated Deep Learning

Federated learning (FL) was proposed by Google as a workaround to utilize privacy-related data in training machine learning models, without intruding the plain text data (Konečnỳ et al., 2016). Over the recent years, the federated learning architecture has been formalized into two categories, namely vertical and horizontal federated learning (Yang et al., 2019b). Both categories of federated learning have great potential in various domains, including user-computer interaction (Phong et al., 2018), medical image analysis (Sheller et al., 2018), financial data analysis (Yang et al., 2019b,a; He et al., 2020) and many more.

To our knowledge, existing federated learning applications mainly adopted machine learning techniques, such as logistic regression and XGBoost, rather than deep neural networks (DNNs). When training a model on federated frameworks, convergence is substantially slower than training on a regular platform. At the end of each training round, gradients and model updates need to be encrypted and transferred to respective recipients, who then decrypts the contents and apply the model updates. For DNNs, the number of trainable parameters and required dataset size are at a totally different scale. Without sufficient support in hardware acceleration, these efficiency obstacles might prove infeasible in DNN training.

A mere example of federated DNN training is found in (Sheller et al., 2018), where a U-Net segmentation model is trained on the BraTS dataset. The authors compared the segmentation accuracy of models trained with centralized data, FL, and institutional incremental learning. However, data security measure was not mentioned in their paper. Comparison of training efficiency is also missing from the report.

To comprehensively evaluate a federated learning system, we must incorporate three key criteria, namely, time efficiency, data security, and model performance. In this paper, we are going to show that these criteria contradict with each other. An optimal design should reach balanced decision among the three.

2.2 Differential Privacy

Differential privacy (Dwork et al., 2006, 2014) is defined in terms of the statistical behaviour of a random process on adjacent datasets. Two datasets

are said to be adjacent if they differ in only one entry. Then, a randomized mechanism $\mathcal{F} : \mathcal{D} \rightarrow \mathcal{R}$ is defined (ϵ, δ)-differentially private, if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$ it holds that

$$\Pr\left[\mathcal{F}(d) \in \mathcal{S}\right] \leq e^{\epsilon} \Pr\left[\mathcal{F}(d') \in \mathcal{S}\right] + \delta. \quad (1)$$

For a deterministic process $\mathcal{M}$ of inputs $d \in \mathcal{D}$, it is common practice to add noise upon its outputs to ensure differential privacy, i.e.,

$$\mathcal{F}(d) = \mathcal{M}(d) + \mathcal{N}(\not{r}, \sigma^2 \cdot S_{\mathcal{M}}) \quad (2)$$

Abadi et al. formalized the application of differential privacy in deep learning (Abadi et al., 2016). The authors also derived a tighter upper bound for the iteratively accumulated ϵ and δ over the training process. It was shown that the level of ϵ and δ spent at each iteration is related to the sampling ratio and the noise level.

Abadi et al.'s work has been extended into federated learning at the client level (Geyer et al., 2017; McMahan et al., 2017). Clients were randomly subsampled to participate in the t-th round of federated model update. The (ϵ, δ) spent were accounted on the central server, where client updates are gathered and aggregated. There are two problems in this process. First, the client datasets are not protected from the central server by differential privacy. Second, the (ϵ, δ) spent at each client is not accounted for individually, while they might differ drastically when their dataset sizes vary.

In this paper, we re-formulate the differentially private federated learning process, so that client dataset privacy is protected from the central server and each other. Noise is added to the accumulated local updates at client servers, before they are sent to the central server. Also, the (ϵ, δ) consumption is accounted for each client respectively, so that the desired privacy level would be protected regardless of its dataset size.

2.3 Additively Homomorphic Encryption

Additively homomorphic encryption (HE) (Gentry et al., 2009; Brakerski and Vaikuntanathan, 2014) provides a way of differential privacy between central server and clients in federated learning (Phong et al., 2018). In the process proposed by Phong et al., client parameter updates are encrypted with a secret key held by the clients only. When the central server receives updates from all clients, it performs additive aggregation without decrypting the gradients. The aggregated model parameters are sent back to client servers, where each client decrypts the contents using the private key. The additive homomorphism enables the aggregation without decryption. In this process, differential privacy and communication security are achieved in a single lossless encryption process (Hardy et al., 2017). However, because the private key must be shared among all clients, it does not guarantee differential privacy between clients. Especially, when there are only two participating clients, one can easily acquire the gradients of another from the decrypted aggregation.

2.4 Sentence-level Text Intent Classification

Intent classification (Li et al., 2008) is one of the fundamental problems in natural language processing. It is crucial for applications such as smart customer service, review categorization, etc.

State of the art text intent classification studies mostly employed deep neural networks (Yin and Schütze, 2016). Common practice in these networks is to represent words in the lexicon with embedding vectors, followed by convolutional or recurrent network modules to extract sentence-level features. TextCNN (Kim, 2014) adapted the convolutional network structure from computer vision domain to tackle the sentence-level classification problem. Zhang et al. evaluated the performance of TextCNN with a wide range of convolutional configurations on public datasets such as MR, SST, TREC, etc. (Zhang and Wallace, 2015). Recently, large recurrent networks further improved accuracies in various natural language processing (NLP) tasks, including text classification (Devlin et al., 2018; Yang et al., 2019c). These models rely on powerful computation resources and large scale of data in training. Once pre-trained, they can be fine-tuned for numerous NLP tasks with smaller datasets. Most of the state of the art text classification accuracies on public datasets originate from these models nowadays. However, these models are often too large to fit in common GPUs with only 16GB graphic memory of even less. Therefore, we opt to carry out experiments on text classification with the simple but efficient TextCNN structure. Results are compared with the baseline accuracies provided in (Zhang and Wallace, 2015).

3 Proposed Methods

3.1 Differentially Privacy Accountant

Federated TextCNN is implemented on the Federated Average platform with distributed training servers.

On a deep learning framework, to protect the privacy of local datasets, Gaussian noise is added to the gradients before they were applied in parameter updates. By the arguments in (Abadi et al., 2016), given δ and noise multiplier σ, an upper bound of the privacy loss ϵ can be computed from

$$\epsilon_t = A(t, q, \sigma, \delta), \tag{3}$$

where $A(\cdot)$ is implemented with numerical integration according to (Abadi et al., 2016). During training, a privacy accountant keeps track of the spent ϵ_t. Once the cumulated ϵ_t exceeds the predefined level, the training must stop sampling from the dataset. Otherwise, privacy of the dataset is considered violated.

3.2 Institutional Differentially Private Federated Training

For federated model training, the gradients computed from local datasets are communicated to the central server at the end of each epoch. That is, given a the current parameter values on the central server as the starting point, client servers sample their dataset batch by batch. For each sample batch, gradients are computed and applied to update the local parameters. After iterating over all batches, the cumulated difference of parameter values is to be sent to the central server for cross-client aggregation. Assuming that the communication channels between the central server and clients are encrypted and safe from interception, the only adversary that might affect the client dataset security comes from the central server itself. It was proven that, given gradients of a convolutional network, it is possible to deduce the actual contents of the input images (Phong et al., 2018). In this paper, we adapt the privacy preservation scheme proposed by Abadi et al. to the federated training procedure, in order to protect client datasets from probing by the central server.

The pseudo codes of the proposed differentially-private federated training procedure is depicted in Algorithm 1.

Algorithm 1 Federated Learning with Differential Privacy

$\mathcal{D} = \{\mathcal{D}_1, ..., \mathcal{D}_K\}$: datasets held by clients $1, ..., K$
$\mathcal{L}$: target loss function
$\mathbf{\Theta}$: trainable parameters
C: gradient norm bound
η: step size
procedure FEDERATEDTRAIN
  Initialize $\mathbf{\Theta}^{(0)}$
  for $t \in \{1, ..., T\}$ **do**
    for all $\mathcal{D}_k$ **do**
      $\Delta_k^{(t)} \leftarrow \text{ClientUpdate}(\mathbf{\Theta}^{(t-1)}, \mathcal{D}_k)$
    end for
    $\Delta^{(t)} \leftarrow \frac{1}{K} \sum_k \Delta_k^{(t)}$
    $\mathbf{\Theta}^{(t)} \leftarrow \mathbf{\Theta}^{(t-1)} + \Delta^{(t)}$
  end for
end procedure
function CLIENTUPDATE($\mathbf{\Theta}_0$, d)
  L: lot size
  t: the number of samples drawn from this dataset
  E: Maximum allowed privacy cost
  $\epsilon \leftarrow \text{PrivacyAccountant}(\sigma, L/|d|, t)$
  if $\epsilon \geq E$ **then**
    return 0
  end if
  $\mathbf{\Theta} \leftarrow \mathbf{\Theta}_0$
  Lot $\mathcal{L} \leftarrow L$ samples from d
  Batches $\{\mathcal{B}_1, ..., \mathcal{B}_B\} \leftarrow$ Random batches of $\mathcal{L}$
  for b in $1, ..., B$ **do**
    $\mathbf{g} \leftarrow \nabla_{\mathbf{\Theta}} \mathcal{L}(\mathbf{\Theta}, \mathcal{B}_b)$
    $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \eta \mathbf{g}$
  end for
  $\Delta_{\mathbf{\Theta}} \leftarrow \text{ClipNorm}(\mathbf{\Theta} - \mathbf{\Theta}_0, C)$
  $\Delta_{\mathbf{\Theta}} \leftarrow \Delta_{\mathbf{\Theta}} + \mathcal{N}(\mathbf{0}, \sigma^2 C^2 \mathbf{I})$
  $t \leftarrow t + 1$
  return $\Delta_{\mathbf{\Theta}}$
end function

The proposed procedure differs from previous differentially private federated learning in the following aspects:

- The proposed procedure protects per-sample privacy of each participating client dataset, instead of the client-level privacy as defined in (McMahan et al., 2017). The proposed procedure is coherent with institutional federated learning applications, where the number of participating datasets is small (usually smaller

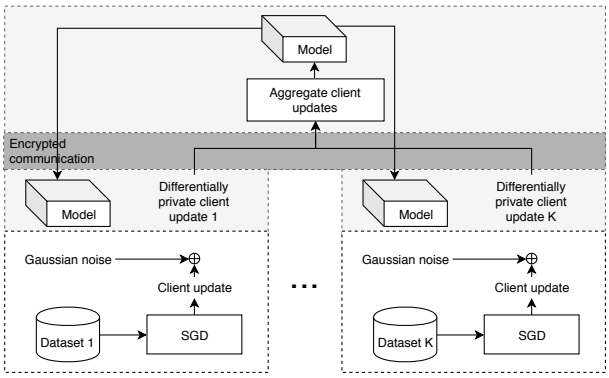


Figure 1: Architecture of a differentially private federated learning system. The shades of the background indicates different security boundaries.

than 5), while the size of each dataset is large. Sampling of clients is prohibitive in this case.

- Differential privacy accountant is performed by clients individually. In institutional FL, the dataset sizes can be very different among the clients. Depending on the setting of sampling strategy, the sampling ratio q can be different from client to client. Thus the privacy loss is variant by the client dataset. Each client should keep record of the spent privacy along each update, and stops its update to central server once the predefined privacy threshold is reached.

- Security distinction between client and central server is made clear. In the proposed procedure, the central server is assumed to be honest-but-curious. Clients can trust the broadcasts from the central server, but should not expose unprotected information to it. Therefore, the noise adding and differential privacy accountant on the transferred gradients are performed on the client side, instead of on the central server.

  The security boundaries are further depicted in Figure 1. In this figure, the light gray areas represent the information shared among the clients and the central server, therefore must be protected by differential privacy. The dark gray area stands for the communication of critical information exposed to not only participants of the FL procedure, but also to third party interception, that must be protected by cryptology.

3.3 Handling Imbalanced Data Load

As mentioned in Section 3.2, data load imbalance is one of the critical considerations in institutional federated learning. It is not uncommon to have several times difference among dataset sizes. In such cases, a number of issues would affect the performance of the federated model.

When differential privacy is involved, the training schedule on each dataset must conform to the predefined privacy limit. The number of samples that can be drawn from a dataset without violating the privacy limit is co-variant to the privacy limit E, sampling ratio q and the noise multiplier σ. A straight-forward solution is to apply the same lot size and noise multiplier over all client datasets. If the privacy accounted has reached predefined limit, the client would stop sampling from its dataset and return zeros for parameter updates. In case of severe data load imbalance, some datasets may stop contributing to the federated model training at an early stage, causing the learned model to be biased towards datasets on other clients.

In our experiments, optimal settings of q and σ are selected according to simulated experiments on balanced datasets. Given the privacy threshold E and the desired training epoch E, series of experiments are conducted to verify the test accuracies using different combinations of q and σ.

4 Experiments

4.1 Implementation Details

TextCNN is a convolutional neural network designed for sentence-level classification tasks (Kim, 2014). It is one of the fundamental structures in the natural language processing (NLP) community. In TextCNN, words are represented by embedding vectors. The word embeddings can be pre-trained from separate datasets, or trained from scratch in an end-to-end fashion. Convolutional layers with various filter widths and feature maps extract features from the concatenated word embeddings in a context-aware manner. Then, max-over-time pooling is performed to aggregate the features into a fixed length vector. A fully connected layer with softmax activation translates the feature vector into sentence classification results.

The CNN structure with the best accuracy on TREC dataset (of Standards and Technology, 2019) is adopted in our implementation. Specifically, 4 convolution layers with filter region sizes 2, 3, 4, 5

respectively are chained sequentially. Each convolution layer has feature depth 400.

Parameters in the model are optimized with respect to the cross entropy loss of the classification outputs, i.e.

$$\mathcal{L}(\mathcal{D};\mathbf{\Theta}) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{i=1}^{M} y_i \log p_i(\mathbf{x};\mathbf{\Theta}), \quad (4)$$

where $\mathbf{x}$ is the input sequence, $\mathbf{y} = \{y_1, ..., y_M\}$ is the one-hot intent class label, and $p_i(\cdot)$'s are the predicted probability of entry $\mathbf{x}$ being class i.

In our experiments, the federated training process is implemented with the coMind collaborative machine learning framework (Roman, 2019). The coMind framework supports distributed GPU training with a federated averaging optimizer. We simulate multi-institution settings within a local area network (LAN), with a central server and 1 to 4 client machines. RSA encryption is used to protect the communication between clients and central server. Each client machine is equipped with an NVIDIA P100 GPU with 16GB graphic memory.

The TextCNN model is implemented on TensorFlow. On client updates, model parameters are trained with the Adam optimizer (Kingma and Ba, 2014), with initial learning rate 0.001. In the reported experiments, we fix the batch size to 64, while the lot size varies with respective experiment settings. During optimization, we use a 0.5 dropout probability to improve model generality.

4.2 Dataset

The TREC dataset (of Standards and Technology, 2019) is a public dataset of NLP text materials. TREC question dataset task involves classifying a question into 6 question types (whether the question is about person, location, numeric information, etc.). This data collection contains all the data used in learning question classification experiment, which has question class definition. The total Dataset size is 5,952, train set size is 5,452, test size is 500. The average length is 10, maximum length is 38. The Vocabulary size is 9,592.

4.3 Results

4.3.1 Baseline

Figure 2 illustrates the training curves of TextCNN on centralized TREC dataset. As the figure shows, the model converges after around 200 iterations. Test accuracy slightly increases over the 200 to 500 iterations. The best test accuracy is 91.2% in our experiment, coherent with the results reported in (Zhang and Wallace, 2015). This experiment serves as the baseline of all following experiments.

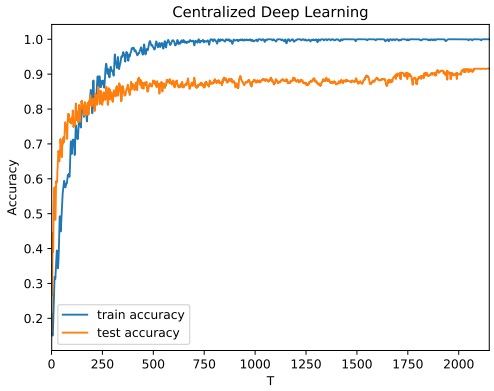


Figure 2: Training and testing accuracies over the training iterations on centralized TextCNN model.

4.3.2 FL with balanced data load

Experiments on federated learning with balanced data loads are performed to evaluate the effect of various hyper parameters in the differentially private FL. In these experiments, the TREC training set is split into $K = 2, 3, 4$ clients with equal number of samples. Without differential privacy, the baseline accuracies of the federated TextCNN models are reported in Table 1. As the number of clients increases, the test accuracy of the federated model decreases. When the training set is divided into 4 clients, the accuracy has dropped by 4.8% compared to the centralized model. Figure 3 illustrates the convergence curves for non-differentially private FL on 2 to 4 clients. The max number of epochs is set to 50 and the batch size is set to 64. Model averaging is performed for every 2 local batch update. Because the dataset sizes are smaller when the number of clients is larger, the number of communication rounds (CR) is also smaller given the same epoch. We can see from Figure 3 that the convergence rates with 2 to 4 clients are similar with each other. The test accuracy on 2 clients slightly improves after 750 CRs, when the 3 and 4 client training has stopped because the maximum epoch has been reached.

On this basis, we would like to study the effect of hyper parameters in differential privacy. The privacy spent ϵ_t is tracked at each communication round between the client and the server. If the privacy accounted has reached predefined limit, the client would stop uploading any updates to the

Model	# of Clients	Test Accuracy
Centralized	1	91.2%
Federated	2	90.0%
	3	87.8%
	4	86.4%

Table 1: Baseline test accuracies of TextCNN without differential privacy.

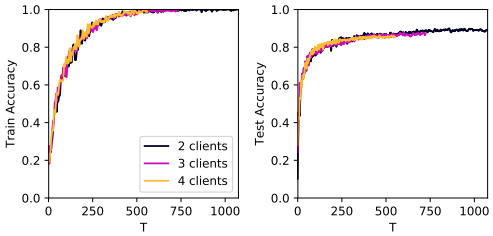


Figure 3: Training and testing accuracies versus communication rounds without differential privacy.

server. When all clients stop updating, the training would terminated before reaching the predefined epoch.

Firstly, suppose that the training set is equally split into 3 clients, given $H = 50$, $\sigma = 4$, lot size $L = 128$ and batch size $B = 64$, the training procedure under different E tolerance is demonstrated in Figure 4. It is shown that the maximally allowd ϵ decides the length of training procedure. For $E = 1$ and $E = 2$, the training only continued for 117 and 476 communication rounds, respectively, equivalent to epochs 8.24 and 33.52. The models are clearly not converged. The resulting test accuracies are thus significantly lower than the baseline.

Secondly, given $H = 50$ and $E = 4$, we would like to see how the lot size L and the noise multiplier σ affect the test accuracies. Again, the TREC training set is equally split into 2 to 4 clients. Tables 2 and 3 shows the test accuracies when varying L and σ respectively. When the noise multiplier σ varies from 2.0 to 8.0, we do not observe a significant difference in the test accuracies. In some cases, the test accuracy may even be slightly higher when σ is large. In contrast, varying lot size L

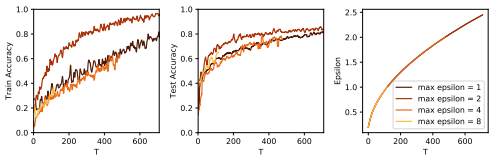


Figure 4: Training and testing accuracies versus communication rounds with varying E.

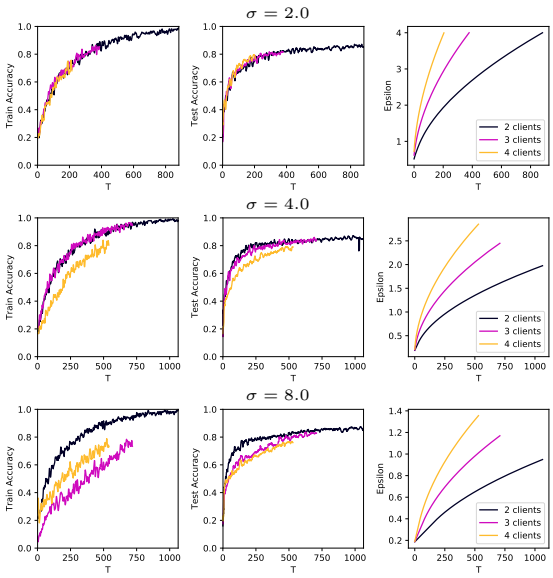


Figure 5: Training and testing accuracies versus communication rounds with varying σ.

K \ σ	2.0	4.0	8.0
2	87.2%	87.2%	87.6%
3	81.8%	84.0%	83.6%
4	79.6%	79.4%	76.8%

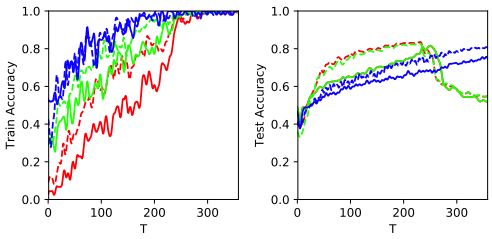
Table 2: Test accuracies of differentially private federated TextCNN models with $L = 128$ and varying σ.

while fixing σ has a large impact on the test accuracy. When the lot size is too large, the model cannot be trained sufficiently before ϵ_t exceeds the predefined threshold E. Yet, in the range where sufficient communication rounds can be performed, a larger lot size gives better performance. Therefore, in the following experiments, we plan the schedule of differentially private federated learning by selecting σ and q according to given privacy tolerance E and training epoch H.

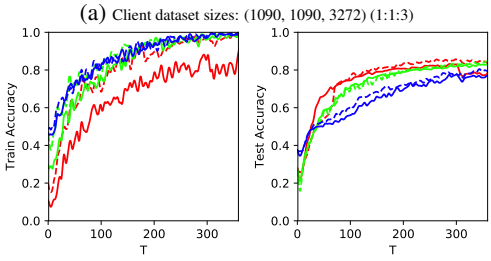
Training curves of varying σ under the same E constraint is also depicted in Figure 5.

L (q)	128 (0.07)	256 (0.14)	512 (0.28)	1024 (0.56)
Acc	84.0%	88.2%	50.2%	45.6%
CR	354	354	108	27

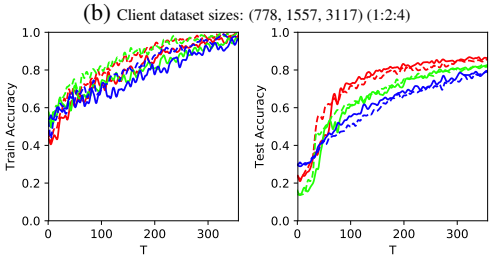
Table 3: Test accuracies of differentially private federated TextCNN models with fixed $\sigma = 4.0$ and varying L.



(a) Client dataset sizes: (1090, 1090, 3272) (1:1:3)



(b) Client dataset sizes: (778, 1557, 3117) (1:2:4)



(c) Client dataset sizes: (1817, 1817, 1818) (1:1:1)

Figure 6: Training and testing accuracies versus training rounds on different data distributions.

4.3.3 FL with imbalanced data load

Unlike business-to-client applications, dataset sizes in institutional FL can be very different from client to client. Given $H = 50$ and $E = 4.0$, $q = 0.14$, $\sigma = 4.0$ are selected by their performance on the balanced datasets. The performance of FL on different data distributions is illustrated in Figure 6, where red (gray), blue (dark), and green (light) colors stand for Client 1, 2, 3, respectively. The dashed lines are statistics of the non-differentially private FL models. The solid lines are from differentially private FL models. The TREC dataset is split into 3 clients with different proportions. We selected the $1:1:3$ and $1:2:4$ ratio to compare with the euqal distribution (e.g. $1:1:1$). The $1:1:3$ ratio can represent the one client having dominant size dataset size over others. The $1:2:4$ ratio can represents each clients having diverse size of dataset. Training and testing accuracies versus training rounds are depicted for each client. The training accuracy on smaller datasets reaches 1.0 shortly after training starts (red lines on Figures 6a and 6b). The larger datasets, however, takes longer to converge. Over the iterations, the training accuracies on different datasets fluctuate a lot when data distribution is highly imbalanced (Figure 6b). The test accuracy directly before model averaging also changes from iteration to iteration. In case where one dataset takes the dominant proportion of data, the fluctuation in test accuracy is less obvious (Figure 6a).

Figure 6 also shows the non-differentially private FL performances along with the differentially private counterparts. At convergence, the training and testing accuracies do not have a large difference between the differentially private and non-differentially private models.

5 Conclusion

Federated learning provides a promising platform for institutions to cooperate with each other in model training, without tampering their data security. Unlike previous works that focus on client-level privacy, this paper addresses the privacy protection issues on the sample-level, which is more appropriate for institutional federated learning. In the proposed procedure, (ϵ, δ)-differential privacy is applied to protect client information from probing by other FL participants. A classical NLP algorithm, TextCNN, is implemented on the differentially private FL platform. Extensive experiments show that, the sampling ratio has large impact on the performance of the FL models. On the other hand, the differentially private FL training is robust to different noise multiplier levels. To address the imbalanced data load situations commonly seen in institutional FL problems, extensive experiments are also conducted to evaluate its influence. Compared with equally sized client datasets, the FL models trained on imbalanced clients see significant decline in test accuracies. Future studies could be devoted to improving the model performance with unequally sized client datasets.

NLP has received a lot of attention from both the academic and commercial societies. For applications in automated banking services, insurance inquires, etc., existing datasets are often confined within the internal servers of respective institutions. There has been ongoing demands on secure ways to utilize these separated data in training a universal model for NLP tasks such as text understanding, question answering, etc. The formulation of institutional federated learning procedure would accelerate development in these areas.

Acknowledgement

This paper is supported by National Key Research and Development Program of China under grant No. 2018YFB1003500, No. 2018YFB0204400 and No. 2017YFB1401202. Corresponding author is Jianzong Wang (jzwang@188.com) from Ping An Technology (Shenzhen) Co., Ltd.

References

Martin Abadi, Andy Chu, and et al. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM.

Zvika Brakerski and Vinod Vaikuntanathan. 2014. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on Computing*, 43(2):831–871.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Cynthia Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Craig Gentry et al. 2009. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178.

Robin C Geyer, Tassilo Klein, and Moin Nabi. 2017. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*.

Anxun He, Jianzong Wang, Zhangcheng Huang, and Jing Xiao. 2020. Fedsmart: An auto updating federated learning optimization mechanism. *arXiv preprint arXiv:2009.07455*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

Lingwei Kong, Hengtao Tao, Jianzong Wang, zhangcheng Huang, and Jing Xiao. 2020. Network coding for federated learning systems. In *International Conference on Neural Information Processing*. Springer.

Xiao Li, Ye-Yi Wang, and Alex Acero. 2008. Learning query intent from regularized click graphs. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 339–346. ACM.

W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y. Liang, Q. Yang, D. Niyato, and C. Miao. 2020. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.

Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345.

Alberto Roman. 2019. comind collaborative machine learning framework.

Micah J Sheller, G Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2018. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, pages 92–104. Springer.

National Institute of Standards and Technology. 2019. Text retrieval conference (trec) data.

K. Yang, T. Jiang, Y. Shi, and Z. Ding. 2020. Federated learning via over-the-air computation. *IEEE Transactions on Wireless Communications*, 19(3):2022–2035.

Mengwei Yang, Linqi Song, Jie Xu, Congduan Li, and Guozhen Tan. 2019a. The tradeoff between privacy and accuracy in anomaly detection using federated xgboost. *arXiv preprint arXiv:1907.07157*.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019b. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019c. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Wenpeng Yin and Hinrich Schütze. 2016. Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513*.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Xinghua Zhu, Jianzong Wang, Zhenhou Hong, Tian Xia, and Jing Xiao. 2019. Federated learning of unsegmented chinese text recognition model. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1341–1345. IEEE.