A Secure and Efficient Federated Learning Framework for NLP

Jieren Deng[1§], Chenghong Wang[2§], Xianrui Meng[3], Yijue Wang[1], Ji Li[4], Sheng Lin[5]
Shuo Han[6], Fei Miao[1], Sanguthevar Rajasekaran[1], Caiwen Ding[1]
[1]University of Connecticut, [2]Duke University, [3]Facebook, [4]Microsoft
[5]Northeastern University, [6]University of Illinois at Chicago
{jieren.deng,yijue.wang,fei.miao,sanguthevar.rajasekaran,caiwen.ding}@uconn.edu
chenghong.wang552@duke.edu,{xianruimeng,changzhouliji}@gmail.com
lin.sheng@northeastern.edu, hanshuo@uic.edu

Abstract

In this work, we consider the problem of designing secure and efficient federated learning (FL) frameworks. Existing solutions either involve a trusted aggregator or require heavyweight cryptographic primitives, which degrades performance significantly. Moreover, many existing secure FL designs work only under the restrictive assumption that none of the clients can be dropped out from the training protocol. To tackle these problems, we propose SEFL, a secure and efficient FL framework that (1) eliminates the need for the trusted entities; (2) achieves similar and even better model accuracy compared with existing FL designs; (3) is resilient to client dropouts. Through extensive experimental studies on natural language processing (NLP) tasks, we demonstrate that the SEFL achieves comparable accuracy compared to existing FL solutions, and the proposed pruning technique can improve runtime performance up to 13.7×.

1 Introduction

Deep Neural Networks have played a significant role in advancing many applications (Yuan et al., 2021; Ding et al., 2017). The field of Natural Language Processing (NLP) leverages Recurrent Neural Networks (RNNs) and Transformers to achieve outstanding performance on many tasks. The Transformer was first introduced in (Vaswani et al., 2017) using a self-attention mechanism and it achieved prominent performance in various NLP tasks. The benefits of RNNs and Transformers in NLP are well-publicized, but the various privacy and security problems still pose challenges to the utilization of these models by data owners, especially users with sensitive data such as location, health, and financial datasets. *Federated Learning* (FL) (McMahan et al., 2017a) empowers different data owners (e.g., organizations or edge devices) to collaboratively train a model without sharing their own data, thus allowing them to address key issues like data privacy. Although data exchanged in FL consists of less information of the user's raw data (Bonawitz et al., 2019), one might still be concerned about how much information remains. Recent research has shown that attackers can still infer sensitive information about the training data, or even reconstruct the it solely from publicly shared model parameters. (Zhu et al., 2019).

Although a series of works (Bonawitz et al., 2017; Truex et al., 2019; Papernot et al., 2018; Wu et al., 2021; Lin et al., 2020) have been proposed to protect FL protocols from leaking sensitive information (Wang et al., 2021; Deng et al., 2021; Wang et al., 2020). They either have to involve a trusted third party (centralized aggregator), or do not tolerate client dropouts. Therefore, the data owners either need to blindly trust the centralized aggregator or must be online all the time during the training period, which makes the entire design less practical. To address the aforementioned issues, in this work, we develop a secure and efficient FL framework, SEFL. It employs two non-colluding servers, i.e., *Aggregation Server* (AS) and *Cryptography Service Provider* (CSP). AS collects the encrypted local updates from clients, and securely aggregates them, while CSP manages the cryptography primitives, i.e. the decryption key. The overarching goal of this framework is to support accurate and efficient RNN and Transformer training while preserving the privacy of training data against the untrusted servers. In other word, any servers' knowledge about any single training data should be bounded by differential privacy (Dwork, 2008).

Our contributions are summarized as follows: (1) We present a novel secure FL framework that eliminates the need for trusted aggregators. (2) SEFL is more resilient to clients dropping out than previous works. SEFL is able to produce a correct global

[§]Equal contribution, alphabetical order

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7676–7682
November 7–11, 2021. ©2021 Association for Computational Linguistics

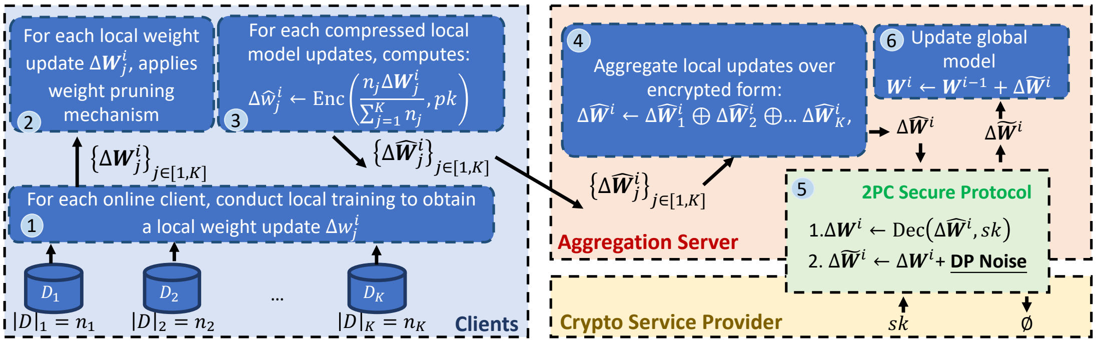

For each local weight update $\Delta \boldsymbol{W}_j^i$, applies weight pruning mechanism
2
For each compressed local model updates, computes:
$\Delta \widehat{w}_j^i \leftarrow \text{Enc}\left(\frac{n_j \Delta \boldsymbol{W}_j^i}{\sum_{j=1}^K n_j}, pk\right)$
3
$\{\Delta \boldsymbol{W}_j^i\}_{j\in[1,K]}$
$\{\Delta \widehat{\boldsymbol{W}}_j^i\}_{j\in[1,K]}$
For each online client, conduct local training to obtain a local weight update Δw_j^i
1
D_1
D_2
...
D_K
$|D|_1 = n_1$
$|D|_2 = n_2$
$|D|_K = n_K$
Clients
4
Aggregate local updates over encrypted form:
$\Delta \widehat{\boldsymbol{W}}^i \leftarrow \Delta \widehat{\boldsymbol{W}}_1^i \oplus \Delta \widehat{\boldsymbol{W}}_2^i \oplus \ldots \Delta \widehat{\boldsymbol{W}}_K^i,$
6
Update global model
$\boldsymbol{W}^i \leftarrow \boldsymbol{W}^{i-1} + \Delta \widetilde{\boldsymbol{W}}^i$
$\Delta \widehat{\boldsymbol{W}}^i$
$\Delta \widetilde{\boldsymbol{W}}^i$
$\{\Delta \widehat{\boldsymbol{W}}_j^i\}_{j\in[1,K]}$
5
2PC Secure Protocol
1.$\Delta \boldsymbol{W}^i \leftarrow \text{Dec}(\Delta \widehat{\boldsymbol{W}}^i, sk)$
2. $\Delta \widetilde{\boldsymbol{W}}^i \leftarrow \Delta \boldsymbol{W}^i$+ DP Noise
Aggregation Server
Crypto Service Provider
sk
$\emptyset$

Figure 1: SEFL workflow

model even 75% of clients are dropped out from the training protocol. (3) To improve the training performance, we integrate the Hankel-matrix based local update/weight pruning method with SEFL to simultaneously reduce the volume of local update and weight storage. The reduction in space, computational, and communication complexity are significant, from O(l^2) to O($2l-1$) for weight/update representation, where l is the block size. With extensive experiments, we show that SEFL achieves comparable or even better accuracy than existing secure FL solutions over complex RNN and Transformer models, and the proposed pruning scheme improves SEFL's performance up to 13.7×.

2 Background

Differential privacy. Let $\epsilon, \delta > 0$ be privacy parameters, a randomized mechanism $\mathcal{M}$ satisfies ϵ, δ-differential privacy (ϵ, δ-DP) if and only if for any two adjacent datasets D and D' (differ by addition or removal of one data), for any possible output S, the following holds:

$$\Pr\left[\mathcal{M}(D) \in S\right] \le e^{\epsilon} Pr\left[\mathcal{M}(D') \in S\right] + \delta$$

The Gaussian Mechanism (GM) (Dwork et al., 2014) achieves differential privacy by approximating a deterministic real-valued function f with an additive noise that is proportional to the function's sensitivity S_f, where $S_f = \max_{D,D'} |f(D) - f(D')|$. A GM is written as $\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 S_f^2)$, where $\mathcal{N}$ denotes a normal distribution, and σ is the noise scale.

Additively homomorphic encryption (AHE). AHE is a semantic secure public-key encryption scheme (Peter et al., 2012), with three algorithms Gen, Enc and Dec, where Gen generates they public and secret key pairs $(\mathsf{pk}, \mathsf{sk})$, Enc encrypts a message with pk and Dec decrypts a ciphertext with secret key sk. In addition AHE provides a homomorphic addition operator $\oplus$, such that $\mathsf{Dec}(\mathsf{Enc}(m_1, \mathsf{pk}) \oplus \mathsf{Enc}(m_2, \mathsf{pk}) \ldots \oplus \mathsf{Enc}(m_k, \mathsf{pk}), \mathsf{sk}) = m_1 + \cdots + m_k$.

Two party secure computation (2PC). 2PC allows two parties with private inputs x_1 and x_2 to jointly compute a given function f. Both parties learn nothing beyond the output of f. A typical 2PC design is the garbled circuit (GC) (Yao, 1986).

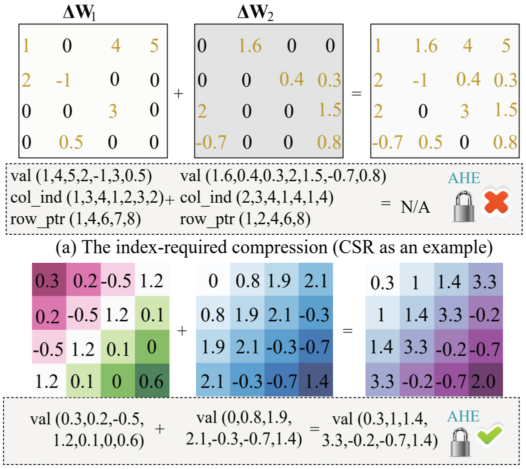
3 SEFL Explained

3.1 Workflow

We design SEFL framework based on the two non-colluding (untrusted) server setting, where an aggregation server (AS) aggregates the encrypted local model updates and another sever (CSP) manages the cryptography primitives (i.e. the decryption key). To ensure the privacy, we require that any server's knowledge about any single training data is bounded by some differential privacy. Figure 1 illustrates an overview of SEFL.

Initially, CSP generates the key pairs $(\mathsf{pk}, \mathsf{sk})$, stores the secret key sk locally, and broadcasts the public key pk to all other entities (AS and all clients). In our design, CSP is tasked to manage the cryptography primitives (i.e. the sk), thus CSP is the only entity that can decrypt the encrypted messages under the secret key sk. In the meantime, we assume that all entities will agree on a same initial model $\mathbf{W}^0$.

Each training iteration, i.e. i^{th} training round, starts with all clients conduct local training with their respective private data D_j then obtain the local model update $\Delta \mathbf{W}_j^i$. Then, each client prunes the obtained model updates using weight pruning techniques and encrypts the compressed update by computing $\Delta \hat{\mathbf{W}}_j^i \leftarrow \mathsf{Enc}(n_j \Delta \mathbf{W}_j^i / \sum_{j=1}^K n_j, \mathsf{pk})$. Clients then submit the encrypted and compressed updates to the AS.



(a) The index-required compression (CSR as an example)

(b) The BHM format compression

Figure 2: CSR format vs. BHM format

On the server side, AS homomorphically adds all encrypted (pruned) local updates over encrypted form and then obtains $\Delta\hat{\mathbf{W}}^i \leftarrow \Delta\hat{\mathbf{W}}^i_1 \oplus \Delta\hat{\mathbf{W}}^i_2 \oplus ...$. Knowing that $\Delta\hat{\mathbf{W}}^i$ is equal to the encryption of the weighted average of all pruned local updates, that is $\Delta\hat{\mathbf{W}}^i = \mathsf{Enc}(\sum_{j=0}^{K} \frac{n_j \Delta\mathbf{W}^i_j}{\sum_{j=1}^{K}} n_j, pk)$. To decrypt the aggregated global update, AS has to collaborate with CSP, as CSP is the only entity that manages the decryption key. Moreover, sending $\Delta\hat{\mathbf{W}}^i$ directly to CSP for decryption will result in the exact value of $\Delta\mathbf{W}^i$ being exposed to CSP, which violates the privacy guarantee. One possible approach is to have AS homomorphically add some random noise to $\Delta\hat{\mathbf{W}}^i$ and send the distorted global update to CSP for decryption. After receiving the result of decryption, AS removes the noise to obtain the true answer. This prevents CSP from knowing the true value of the global model update, however AS will know this value, which is also a privacy violation. To ensure none of the two servers can learn the exact global updates, in our design, AS first sends a distorted $\Delta\hat{\mathbf{W}}^i$ (with some random mask) to CSP, followed by CSP decrypts the distorted global update. Then, the two servers jointly evaluate a secure 2PC where AS inputs the random mask and CSP inputs the decrypted global update. $\Delta\hat{\mathbf{W}}^i$ is then recovered inside the secure 2PC protocol. Next, each server independently samples a DP noise and provides it as input to the secure 2PC. These DP noises are then added to the recovered $\Delta\hat{\mathbf{W}}^i$ inside the 2PC protocol. Finally, the protocol returns the global update distorted by DP noise to AS, with which AS updates the global model, $\mathbf{W}^i \leftarrow \mathbf{W}^{i-1} + \Delta\tilde{\mathbf{W}}^i$. Note that, the choice of DP noise is quite flexible, and by default, SEFL uses Gaussian noise to distort the global update.

SEFL repeats the training phases until it reaches the maximum training round T or the model is converging. Note that, it is not necessary for AS to have all local updates from clients, according to our evaluation results, SEFL is able to train an accurate model when only 10% of clients contribute their local updates. Therefore, in practice, one can set an aggregation threshold, say L, which means that AS can start aggregating local updates as long as it receives more than L updates.

3.2 Block-Hankel Matrix-based Pruning

Cryptographic primitives can help to provide stronger security guarantees. However, in practice, they often come at high computation and communication overhead. Adding additional cryptographic operations in an FL framework could potentially prohibit the popularity and the adoption of resource-constrained edge devices such as mobile or IoT devices with limited resources (e.g., computation, memory size). Therefore, to be compatible with resource-constrained edge devices on federated learning, we aim to minimize the number of cryptographic operations required during training while maintaining the accuracy of FL. To achieve this, we develop an efficient method to train a large neural network by simultaneously reducing the volume of local updates and weight storage. we design an efficient method to train a large NLP model with reduced volume of local updates, to reduce the number of required cryptographic operations. **Pitfall of sparsity format in AHE.** Typical weight pruning approaches require to store the indices of nonzero entries (Gurevin et al., 2021; Gui et al., 2019; Wen et al., 2016; Ren et al., 2020; Ma et al., 2020). However, the different position of nonzero values from all clients can lead to significant inefficiency for the subsequent model update aggregation. As shown in Fig. 2 (a), assume AS aggregates two local updates with the same sparsity from $\mathcal{C}_1$ and $\mathcal{C}_2$. We apply compressed sparse row (CSR) format, to represent the updates ($\Delta\mathbf{W}_1$ and $\Delta\mathbf{W}_2$ in Fig. 2 (a)), where the non-zero elements of $\Delta\mathbf{W}_1$ and $\Delta\mathbf{W}_2$ are not located in the same position. As the AHE-based update aggregation process is a black-box homomorphic addition operation, we can not reconstruct the original sparse matrix from CSR since indices are encrypted, therefore we can not correctly produce the aggregated update.

Crypto-friendly Block-Hankel matrix based pruning. We divide the local update into multiple modules with identical shape. Within each module, a special format of structure matrix is applied to approximate the original matrix without indices. In our framework, we investigate the use of blocks of Hankel matrix (BHM) to approximate blocks of local update. As shown in Fig. 2 (b), we can perform aggregation based on the encrypted val vectors since the positions of the sequence vectors are identical. In addition, the resultant global model will have the same size, therefore downloading and uploading communication is symmetric and balanced.

In what follows, we discuss the convergence analysis for pruned sub-networks.

Theorem 1. *For every network f with depth l and $\forall\ i \in \{1, 2, \dots, n\}$. Consider g is a randomly initialized neural network with 2n layers, and width $poly(d, n, m)$, where d is input size, n is number of layers in f, m is the maximum number of neurons in a layer. The weight initialization distribution belongs to Uniform distribution in range [-1,1]. Then with probability at least $1 - \beta$ there is a weight-pruned subnetwork $\hat{g}$ such that:*

$$\sup_{x\in\chi, \|W\|\leq 1} \|f(x) - \hat{g}(x)\| \leq \alpha \tag{1}$$

Proof 1 *We start with analysis over simple ReLU networks, where $f(x) = w \cdot x$, $g(x) = \mathbf{u}\sigma(\mathbf{w}^g x)$. Since σ is a ReLU activation function, thus $w = \sigma(w) - \sigma(-w)$ and such that $x^* \mapsto \sigma(wx) = \sigma(\sigma(wx) - \sigma(-wx))$. On the other hand, this neuron can be present as: $x^* \mapsto \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{w}^g x)$. Let $\mathbf{w}^+ = \max\{\mathbf{0}, \mathbf{w}\}$, $\mathbf{w}^- = \min\{\mathbf{0}, \mathbf{w}\}$, $\mathbf{w}^+ + \mathbf{w}^- = \mathbf{w}^g$. Then*

$$x^* \mapsto \mathbf{u}\sigma\left(\sigma\left(\mathbf{p} \odot \mathbf{w}^+ x\right) - \sigma\left(\mathbf{p} \odot -\mathbf{w}^- x\right)\right) \tag{2}$$

Base on (Lueker, 1998), when $n \geq C \log \frac{4}{\alpha}$, $\forall w^f \in [0, 1]$, there exist a pattern of $\mathbf{w}$ and $p \in \{0, 1\}^n$:

$$\Pr\left[\left|w^f - \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{w}^+)\right| < \frac{\alpha}{2}\right] \geq 1 - \frac{\beta}{2} \tag{3}$$

By symmetric, Eq. 3 holds for $\mathbf{w}^-$ as well. Therefore, we obtains sup $\left|w^f x - \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{w}x)\right| \geq \alpha$. To extend it to a single network layer, we computes

$$\begin{aligned} &sup\left|\mathbf{W}^f\mathbf{x} - \mathbf{u}\sigma(\mathbf{p} \odot \mathbf{W}^g\mathbf{x})\right| \\ &\leq \sum_{j=1}^{k}\sum_{i=1}^{m} sup\left|w^f_{j,i}x_i - \mathbf{u}_i\sigma(\mathbf{p}_{j,i} \odot \mathbf{w}_{j,i}x_i)\right| \leq \alpha \end{aligned} \tag{4}$$

We now provide the general case analysis. With probability over $1 - \beta$, we obtains:

$$\begin{aligned} &\|f(x) - \hat{g}(x)\| \\ =&\left\|\mathbf{W}_n\mathbf{x}_n - \mathbf{P}_{2n} \odot \mathbf{W}^g_{2n}\mathbf{x}^g_n\sigma(\mathbf{P}_{2n-1} \odot \mathbf{x}^g_{2n-1})\right\| \\ \leq&\alpha/2 + \alpha/2 = \alpha \end{aligned} \tag{5}$$

Putting it all together. Our objective is to compress the weights and updates using the BHM formats. Thus we minimize the loss function subject to constraints of BHM. More specifically, we set constraints as $\mathbf{S}_i^{(t)} = \{\mathbf{W}_i^{(t)} \mid \mathbf{W}_i^{(t)} \in \text{BHM}\}$. The *backward propagation* process of the training phase can also be implemented using the BHM format, since pruning based on the block Hankel matrix has the same "effectiveness" as unpruned DNNs, as shown in (Zhao et al., 2017).

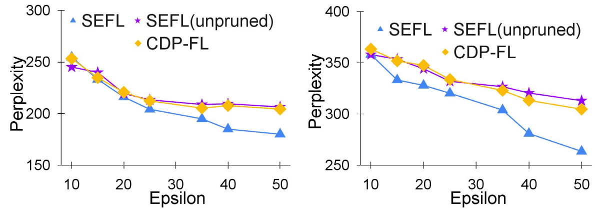
Compared to other index-required pruning methods, the BHM pruning has the following advantages. First, it always guarantees the strong structure of the trained network, thereby avoiding the storage space, computation, and communication time overhead incurred by the complicated indexing process. Second, during training, the BHM-based approach directly trains weight matrices in the BHM format by updating only one vector for each block (i.e., $2l - 1$ vs. l^2). Third, the reduction in space, computational, and communication complexity by using BHM are significant. The weight tensor $\mathbf{W}_i^{(t)}$ and updates $\Delta\mathbf{W}_i^{(t)}$ have the storage complexity and communication complexity reduced from O(l^2) to or O($2l - 1$).

4 Experiments

We implement the SEFL system using PyTorch 1.4.0, CUDA 10.1. All experiments are performed on the AWS EC2 cloud instance with a 2.30GHz Intel Xeon Gold 5218 Salable Processors and 8 NVIDIA Quadro RTX 6000 GPUs. We evaluate SEFL by conducting experiments using LSTM and Transformer on WikiText-2 (Merity et al., 2016) dataset. The LSTM model is adopted from (Hochreiter and Schmidhuber, 1997). The Transformer model (Vaswani et al., 2017) contains two layers with an embedding dimension of 200, two attention heads, and 200 hidden units. We use perplexity to measure the quality of the predicted data for both Transformer and LSTM.

4.1 Result Analysis

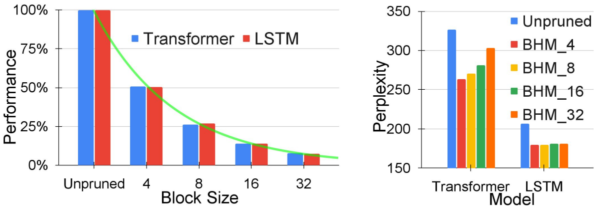
Comparisons with existing private FL. In Figure 3, we compare SEFL with the state-of-art private FL design, CDP-FL (Geyer et al., 2017). First,



(a). LSTM

(b). Transformer

Figure 3: Comparison of the impact of different federated learning approaches on accuracy.



(a) Runtime Performance

(b) Accuracy

Figure 4: Comparison of the impact of different Block Size on accuracy and performance

the unpruned SEFL achieves similar accuracy compared to the CDP-FL due to the fact that the logical training process of these two approaches are similar. Both methods first utilize FedAvg (McMahan et al., 2017b) to obtain a global aggregation model and then distort it with DP noise. The optimized (with pruning technique) SEFL improves the accuracy by up to 11% and 15% over the CDP-FL method, with LSTM and Transformer model, respectively. One possible explanation is that pruning reduces the DP noises added to the aggregation model. Since, to distort the model, one should inject Gaussian noise to each model element independently. Therefore, the smaller the size of the model, the fewer times the Gaussian noise is injected.

Evaluating optimization. We compare the SEFL (with pruning optimization) with unoptimized SEFL in Figure 4. We report the average elapse time in seconds over 10 replicated runs as the runtime performance. We report the accuracy and runtime performance for unpruned SEFL and SEFL with BHM block size from 4 to 32. Note that the larger the block size, the smaller the compressed model will be. SEFL achieves a performance improvement of up to 13.7× over the unpruned SEFL under both LSTM and Transformer models. Additionally, SEFL shows better accuracy (smaller perplexity) compared to the unpruned SEFL in almost all test groups. For best cases, SEFL achieves 19.3%, and 12.8% accuracy improvement,respectively, in contrast to the unoptimized SEFL implementation. BHM-based pruning optimization not only brings significant performance improvements, but also optimizes the accuracy guarantees.

Dropout Rate	BHM (75%)	BHM (50%)	BHM (25%)	BHM (0%)
LSTM	194.82	187.64	178.45	177.41
Transformer	310.12	301.04	279.81	263.78

Table 1: Comparison of the impact of dropout rate.

SEFL with clients dropout. We evaluate whether SEFL is able to handle clients dropouts in Table 1, where we report accuracy when 25%, 50%, and 75% of clients are dropped out from the protocol. Shown in Table 1, when the dropout rate is relatively small, i.e., 25%, SEFL achieves almost the same accuracy guarantee as in the no-dropout case (0.5% and 5% accuracy degradation for LSTM and Transformer, respectively). Even when majority of clients are drooped out, i.e. 75% drop rate, SEFL still produces accurate models with only 17.41 and 46.34 higher perplexity. In summary, SEFL can handle a large number of client dropouts with relatively small degradation in accuracy. This result shows that our proposed approach is applicable to practical scenarios.

5 Conclusion

In this paper, we introduced a new secure and efficient FL framework, SEFL, that (i) eliminates the need for the trusted entities, (ii) achieves similar model accuracy compared with existing FL approaches, and (iii) is resilient to client dropouts. We also proposed optimizations that mitigate the high computation and communication overhead caused by cryptographic primitives. This is achieved by applying a local weight pruning technique based on the block Hankel-matrix. Through extensive experimental studies on NLP tasks, we demonstrate that the SEFL achieves comparable accuracy compared to existing FL solutions, and can significantly improve runtime performance.

6 Acknowledgements

This research was supported in part by UConn REP award (KFS: 4648460), the National Science Foundation (NSF) Grants 1743418, NSF 1843025, NSF 1849246 and NSF 1952096. This research is based upon work supported by the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

References

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *ACM CCS*, pages 1175–1191.

Jieren Deng, Yijue Wang, Ji Li, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding, editors. 2021. *Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics.

Caiwen Ding, Siyu Liao, Yanzhi Wang, Zhe Li, Ning Liu, Youwei Zhuo, Chao Wang, Xuehai Qian, Yu Bai, Geng Yuan, Xiaolong Ma, Yipeng Zhang, Jian Tang, Qinru Qiu, Xue Lin, and Bo Yuan. 2017. Circnn: Accelerating and compressing deep neural networks using block-circulant weight matrices. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-50 '17, page 395–408, New York, NY, USA. Association for Computing Machinery.

Cynthia Dwork. 2008. Differential privacy: A survey of results. In *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation*, TAMC'08, page 1–19, Berlin, Heidelberg. Springer-Verlag.

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.

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

Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. 2019. Model compression with adversarial robustness: A unified optimization framework. In *NeurIPS*, pages 1285–1296.

Deniz Gurevin, Mikhail Bragin, Caiwen Ding, Shanglin Zhou, Lynn Pepin, Bingbing Li, and Fei Miao. 2021. Enabling retrain-free deep neural network pruning using surrogate lagrangian relaxation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2497–2504. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Sheng Lin, Chenghong Wang, Hongjia Li, Jieren Deng, Yanzhi Wang, and Caiwen Ding. 2020. Esmfl: Efficient and secure models for federated learning. *arXiv preprint arXiv:2009.01867*.

George S Lueker. 1998. Exponentially small bounds on the expected optimum of the partition and subset sum problems. *Random Structures & Algorithms*, 12(1):51–62.

Xiaolong Ma, Fu-Ming Guo, Wei Niu, Xue Lin, Jian Tang, Kaisheng Ma, Bin Ren, and Yanzhi Wang. 2020. Pconv: The missing but desirable sparsity in dnn weight pruning for real-time execution on mobile devices. In *AAAI*, pages 5117–5124.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017a. Communication-efficient learning of deep networks from decentralized data. In *(AISTATS)*.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017b. Communication-efficient learning of deep networks from decentralized data.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with PATE. In *ICLR 2018,*.

Andreas Peter, Max Kronberg, Wilke Trei, and Stefan Katzenbeisser. 2012. Additively homomorphic encryption with a double decryption mechanism, revisited. In *ISC*.

Ao Ren, Tao Zhang, Yuhao Wang, Sheng Lin, Peiyan Dong, Yen-Kuang Chen, Yuan Xie, and Yanzhi Wang. 2020. Darb: A density-adaptive regular-block pruning for deep neural networks. In *AAAI*, pages 5495–5502.

Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. 2019. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yijue Wang, Jieren Deng, Dan Guo, Chenghong Wang, Xianrui Meng, Hang Liu, Caiwen Ding, and Sanguthevar Rajasekaran. 2020. Sapag: a self-adaptive privacy attack from gradients. *arXiv preprint arXiv:2009.06228*.

Yijue Wang, Chenghong Wang, Zigeng Wang, Shanglin Zhou, Hang Liu, Jinbo Bi, Caiwen Ding, and Sanguthevar Rajasekaran. 2021. Against membership inference attack: Pruning is all you need. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3141–3147. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082.

Xin Wu, Hao Zheng, Zuochao Dou, Feng Chen, Jieren Deng, Xiang Chen, Shengqian Xu, Guanmin Gao, Mengmeng Li, Zhen Wang, et al. 2021. A novel privacy-preserving federated genome-wide association study framework and its application in identifying potential risk variants in ankylosing spondylitis. *Briefings in Bioinformatics*, 22(3):bbaa090.

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

Geng Yuan, Payman Behnam, Yuxuan Cai, Ali Shafiee, Jingyan Fu, Zhiheng Liao, Zhengang Li, Xiaolong Ma, Jieren Deng, Jinhui Wang, Mahdi Bojnordi, Yanzhi Wang, and Caiwen Ding. 2021. Tinyadc: Peripheral circuit-aware weight pruning framework for mixed-signal dnn accelerators. In *2021 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 926–931.

Liang Zhao, Siyu Liao, Yanzhi Wang, Zhe Li, Jian Tang, and Bo Yuan. 2017. Theoretical properties for neural networks with weight matrices of low displacement rank. In *International Conference on Machine Learning*, pages 4082–4090.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *NeurIPS*, pages 14774–14784.