This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089713, IEEE Internet of Things Journal

Beyond Class-Level Privacy Leakage: Breaking Record-Level Privacy in Federated Learning

Xiaoyong Yuan*, Xiyao Ma†, Lan Zhang*, Yuguang Fang†, *Fellow, IEEE*, Dapeng Wu†, *Fellow, IEEE*
*Michigan Technological University, Houghton, MI 49931, USA
†University of Florida, Gainesville, FL 32611, USA

***Abstract*—Federated learning (FL) enables multiple clients to collaboratively build a global learning model without sharing their own raw data for privacy protection. Unfortunately, recent research still found privacy leakage in FL, especially on image classification tasks, such as the reconstruction of class representatives. Nevertheless, such analysis on image classification tasks is not applicable to uncover the privacy threats against natural language processing (NLP) tasks, whose records composed of sequential texts cannot be grouped as class representatives. The finer (record-level) granularity in NLP tasks not only makes it more challenging to extract individual text records, but also exposes more serious threats. This paper presents the first attempt to explore the record-level privacy leakage against NLP tasks in FL. We propose a framework to investigate the exposure of the records of interest in federated aggregations by leveraging the perplexity of language modeling. Through monitoring the exposure patterns, we propose two correlation attacks to identify the corresponding clients when extracting their specific records. Extensive experimental results demonstrate the effectiveness of the proposed attacks. We have also examined several countermeasures and shown that they are ineffective to mitigate such attacks, and hence further research is expected.**

***Index Terms*—federated learning, language modeling, privacy, neural networks, natural language processing.**

I. Introduction

With the increasing number of large companies compromised on data security and user privacy, federated learning (FL) has recently attracted great attention as a promising privacy-preserving machine learning technique. FL enables collaborative learning among multiple clients (*e.g.*, mobile devices and IoT devices) without sharing their on-device data [1]–[3]. Although the idea appeared in collaborative distributed machine learning [4]–[6] and distributed optimization [7], the concept of FL was first coined by Google, so as to build better language models on the virtual keyboard, Gboard [8]. Federated learning has recently been deployed for multi-institutional collaborations in broader areas, such as medical diagnosis [9], financial fraud detection [10], and the Internet of Things (IoT) in smart homes [11].

Unfortunately recent studies showed that merely keeping the data locally cannot prevent FL from privacy leakage (e.g., membership inference attacks [12], [13], reconstruction attacks [14]–[16]). For example, reconstruction attacks can still infer the class representatives in image classification tasks by analyzing the parameter or gradient updates shared by local models [14]–[16], which raises substantial privacy concerns in federated learning, particularly in healthcare industries.

Fortunately, such class-level image reconstruction attacks [14]–[17] are not applicable to expose privacy threats against federated natural language processing (NLP) tasks. The reconstruction attacks only focus on class representatives, while the training data of natural language tasks are individual records, like the sequential texts, which cannot be grouped by class representatives. This is because one class of the image classification tasks usually consists of a large number of samples, while each sample of NLP tasks represents a unique class. Thus, compared with class-level data privacy, the finer (record-level) granularity in NLP tasks not only challenges the extraction of individual text records, but also exposes more serious threats to leak more precise private information. Besides, the aforementioned privacy attacks usually have stronger assumptions: the federated server is malicious (or at least honest but curious) and has access to the victim client's local model, which is usually impractical in real-world systems. Recent approaches (*e.g.*, secure aggregation [18], [19]) mitigate such attacks by encrypting the local models and reveal only the global model (see Section VI-D). Hence, instead of assuming a malicious federated server, we can consider a more practical scenario where the federated server is trusted, but only some of the clients are compromised. In this situation, the adversary can only access the global model rather than the victim client's local model during federated learning, which increases the difficulty of the aforementioned attacks. One natural question is that under this new assumption, whether there is still privacy concerns in federated NLP learning. Unfortunately, the answer is negative. Our study reveals that we still face two kinds of privacy threats, which is one of the contributions in this paper. Therefore, it is urgent and compelling to carefully investigate privacy threats to federated NLP learning under the new assumption.

Privacy leakage of NLP learning has already been recently investigated in [20] recently, where a record could be extracted through a well-trained neural network. Specifically, a neural network for language modeling memorizes the private records typed by a user during the training. The adversary can extract a private record by auto-completing the record using a well-trained neural network. However, the record extraction approach proposed for standalone NLP models becomes ineffective in FL. Instead of training on records, FL is collaboratively trained by aggregating models that are locally trained by multiple clients. Moreover, FL involves various clients, and thus an adversary can hardly identify the specific client owning the private record, which hinders the adversary

from further attacks (*e.g.*, impersonation attacks). Although federated learning was first proposed for private-preserving NLP tasks [8], the privacy threat of record extraction in NLP has not been thoroughly explored as yet.

In this paper, we propose a unified framework to explore record-level privacy leakage of FL in NLP tasks without the assumption on a malicious federated server. Specifically, due to the distributed learning nature, *e.g.*, the varying communication cost, delay, and computational capability among multiple clients, FL in real-world applications usually uses asynchronous aggregation [21], where clients do not need to update their local models at the same time. Such asynchronous aggregation in FL may cause the imbalanced training of each client during FL. Therefore, by tracking the footprints exposed through imbalance training, we propose two correlation attacks under a unified framework to expose the private records in NLP tasks and reveal the client's identity.

Our main contributions are summarized below.

- This is the first work to explore record-level privacy leakage in federated NLP tasks, potentially revealing client identities. We investigate a practical but more challenging system setting with a trusted server under asynchronous aggregation.
- We propose a unified framework to track the footprints leaked during asynchronous aggregation, analyze the exposure rates of private records, and eventually expose the private records and the client identities.
- We introduce two new correlation attacks in the framework, Eavesdropping attack and Watermark attack. By eavesdropping on the client selection or injecting a single watermark, the adversarial can successfully extract private records and reveal client identities. We evaluate the effectiveness of the attacks on three widely-used datasets.
- We investigate several possible key countermeasures against the proposed attacks, including obfuscating word embedding, avoiding overfitting, and deploying differential privacy. A few countermeasures do reduce the risk of record extraction, but still at the cost of significant performance degradation. This suggests further search for countermeasures is needed.

II. Preliminaries

A. Federated Learning

FL enables collaborative machine learning among multiple clients without sharing their private data [1]–[3]. Suppose N clients jointly train a machine learning model $\mathcal{F}_w : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ denotes the input feature space, and $\mathcal{Y}$ the output space. The FL algorithm can be described as follows:

1) Each client $i \in \{1, 2, \cdots, N\}$ locally collects their data $\mathcal{D}_i$, and initializes a local model $\mathcal{F}_{w_i}$, where w_i denotes the parameters of the local model of Client i;
2) Each client $i \in \{1, 2, \cdots, N\}$ trains a local model $\mathcal{F}_{w_i}$ based on its local dataset $\mathcal{D}_i$ for P epochs (local epochs);
3) Server selects M clients ($M < N$) to upload their local models for asynchronous aggregation;
4) Server aggregates M local models and updates the FL model $\mathcal{F}_w$;
5) Server sends the FL model $\mathcal{F}_w$ back to the N clients, and clients update their local models;
6) Repeat Step 2)-5) for T rounds.

This paper considers one of the leading federated learning algorithms, FedAvg [1], which averages the parameters among all the selected models:

$$w = \frac{\sum_{i \in S} |D_i| w_i}{\sum_{i \in S} |D_i|}, \tag{1}$$

where w denotes the weight of the FL model, $|D_i|$ the size of Client i's data, and S the set of clients. FedAvg has been shown to be convergent on both IID and Non-IID data [22].

B. Language Modeling

In this paper, we target at language modeling tasks [23]–[26]. As one of the typical tasks implementing FL, language modeling is widely applied for commercialized applications, like the auto-completion service used in smartphones. Language modeling takes a sequence of words or characters as input and predicts the next word or character. The probability of the occurrence of a sequential text $\{x_1, \cdots, x_n\}$ can be formulated as:

$$\Pr(x_1, \cdots, x_n) = \Pr(x_1) \Pr(x_2|x_1) \cdots \Pr(x_n|x_1 \cdots x_{n-1}), \tag{2}$$

where x_i is the ith character or word of a sequential text. A language model $\mathcal{F}_w$ with parameters w is trained to maximize the probability of the sequence:

$$\max_w \Pr(x_1, \cdots, x_n|w). \tag{3}$$

With the recent advances in neural networks, Recurrent Neural Network (RNN) and its variants, such as Long Short-Term Memory (LSTM) [27] and Recurrent Highway Network [28], have been applied for language modeling [24].

C. Privacy Threats in Federated Learning

Recent research mainly targets at two types of privacy attacks in federated learning: membership inference attacks and reconstruction attacks. Nasr *et al*. and Melis *et al*. analyzed the membership inference attack against FL [12], [13]: given an FL model, adversaries can infer whether a given data or attribute belongs to the model's training data. For example, Nasr *et al*. showed that federated learning is vulnerable to membership inference attacks [12]. Melis *et al*. found that the property of training datasets can be inferred through the global model [13]. They captured the global model after each aggregation and then trained a classifier to determine if the desired properties belong to the training dataset. For reconstruction attacks, adversaries usually aimed at reconstructing synthetic data samples utilizing Generative Adversarial Networks (GANs) [14]–[17]. For instance, Hitaj *et al*. trained a Generative Adversarial Network (GAN)-based model to infer the class representatives from updated clients' models in federated learning [14]. The parameters of an FL model were converted into a discriminator in the GAN, while the generator was trained to synthesize the representative training samples for each class. Similarly, Wei *et al*. reconstructed the class

representatives by analyzing the parameter updates shared by clients' models [15]. Wang *et al*. extended reconstruction attacks to client-level, where a multi-task discriminator was trained by a GAN model to identify the client of the generated data samples [16]. However, these approaches are not applicable to expose privacy threats against federated natural language processing (NLP) tasks and require access to the victim's local model, which is a strong assumption in practice. In this paper, we demonstrate that the adversary is able to extract individual data records from federated NLP models as well as the client identity even without access to the victim's local model.

III. Threat Model

In this section, we first present the threat model, and then detail the generality and rationale about the adversary.

A. Federated Learning Scenario

Following the FL procedure described in Section II-A, we consider that N clients collaboratively train a language model. The language model learns from clients' private records by aggregating their local models and predicts the next character or word, given a sequence of characters as input. The training data of each client may contain sensitive information that cannot be shared across the clients, *e.g*., identity number, date of birth, and home address. A server aggregates the clients' local models asynchronously without accessing the clients' private data. We assume the server is fully trusted and does not leak clients' local models.

B. Adversary Objective and Capabilities

To explore the record-level privacy in FL, *the objective of the adversary is to extract records of interest and reveal client identities of these records in FL*. We assume the federated server is fully trusted and the adversary could access the global model only. This is the case when one of the clients is compromised or malicious, and the global model can be easily acquired by the adversary. In this paper, the adversary performs in an invisible fashion that the adversary does not change the behavior of the models or affect the integrity (performance) of the models. In this way, the attack is hard to detect through intrusion detection.

It should be emphasized that we do not assume the adversary has access to clients' local models, different from most existing attacks. Existing attacks assume that the server is malicious or honest-but-curious, that is the server (adversary) can at least access the clients' local models and expose client-level privacy information. We argue that if the client local model is exposed to an adversary, many recent attacks against standalone models (*e.g*., model inversion [29], gradient leakage [30], [31]) can be directly used to attack the FL. Therefore, to explore the distinctive threats in FL, we focus on investigating a more challenging situation for the privacy leakage, that is, when the server is trustworthy and follows all protocols in FL, and the adversary has no access to clients' local models.

Under this assumption, the adversary only has access to the parameters of the global model at every global aggregation. In our proposed attacks, we also assume that the adversary knows whether a victim client is selected for each global aggregation (Eavesdropping attack) or can inject a record into the victim client's training data (Watermark attack). We will introduce the details in the following sections.

IV. Record-Level Privacy Attacks

A. A Unified Framework for Record-Level Privacy Attacks

In this section, we propose a unified framework for record-level privacy attacks. In FL, asynchronous aggregation is considered a default and efficient algorithm for model aggregation in practice, which aggregates the weights of models from only a subset of the clients. However, asynchronous aggregation may cause an imbalanced performance of training among clients in each model aggregation. Inspired by this observation, the proposed framework tracks the training footprints of clients exposed during asynchronous aggregation, calculates the exposure rates, analyzes the correlation between the clients and records of interest, and eventually extracts exact text records and reveals client identities. Figure 1 illustrates an overview of the framework for record-level privacy attacks in FL.

We first introduce a measurement of data exposure and show the correlation between data exposure and asynchronous aggregation. Accordingly, we describe two proposed correlation attacks in the framework, namely, Eavesdropping attack and Watermark attack, to explore the privacy risks in FL.

We propose an exposure measurement to assess the risk of record-level leakage. The goal of training an FL model is to memorize the patterns in the training data. However, this may leak sensitive information when extracting data records in the training data. Therefore, we introduce a metric to measure the performance of FL model training using language model perplexity [32]. Then we propose an exposure measurement based on the perplexity.

Definition 1. *Given a sequential text $x_1 \cdots x_n$, the perplexity of a language model [32] is defined as:*

$$\begin{aligned}\text{Perplexity} &\triangleq \Pr(x_1 \cdots x_n)^{-\frac{1}{n}} \\ &= \frac{1}{\sqrt[n]{\prod_{i=1}^{n} \Pr(x_i | x_1, \cdots, x_{i-1})}}. \end{aligned} \tag{4}$$

Perplexity calculates the inverse probability normalized by the number of characters/words (n). A low perplexity indicates the language model is good at predicting the given text. To measure the privacy exposure, we define an exposure rate Exposure, as the negative log perplexity:

Definition 2. *Given a sequential text $x_1 \cdots x_n$, the exposure rate of a language model is defined as:*

$$\begin{aligned}\text{Exposure} &\triangleq -\log \text{Perplexity} \\ &= \frac{1}{n} \sum_{i=1}^{n} \log \Pr(x_i | x_1, \cdots, x_{i-1}). \end{aligned} \tag{5}$$

The exposure rate measures how likely the given text will be exposed (or memorized) through the model.

Based on the definition of exposure rates, we next show the positive correlation between the change of exposure rates

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089713, IEEE Internet of Things Journal

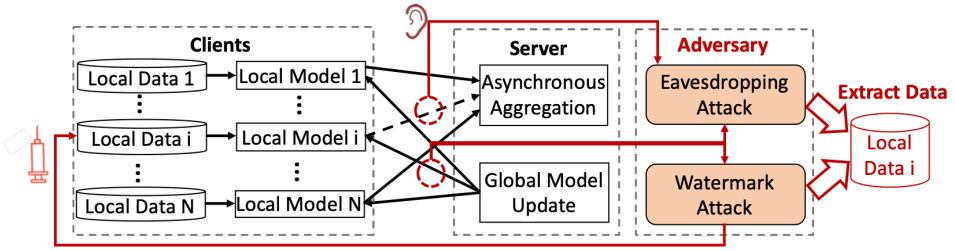


Fig. 1: Overview of the framework for record-level privacy attacks in FL. N clients collaboratively train a global model on their private data through asynchronous aggregation. The adversary snapshot the global model at every aggregation step. To expose the privacy of Client i, Eavesdropping attack eavesdrops the selection of Client i in each asynchronous aggregation. Watermark attack injects a watermark into Client i's local data. By conducting correlation analysis, both attacks can extract the exact private records and reveal the client identities.

and the selections of the victim client in each asynchronous aggregation. We assume that each Client i contains only a record $\boldsymbol{x_i}$. In the tth round of aggregation, each client trains on their local data by minimizing the loss function $\mathcal{L}(w_i, \boldsymbol{x_i})$ using gradient descent:

$$w_i^t = w^{t-1} - \eta \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_i})}{\partial w_i^{t-1}}, \tag{6}$$

where $\mathcal{L}(w_i, \boldsymbol{x_i})$ is the log perplexity and η is the learning rate. w^{t-1} and w_i^{t-1} denote the weight of global model and Client i's local model in the $(t-1)$th round of aggregation, respectively.

Theorem 1. *The change of exposure rates of the victim Client v's private data $\boldsymbol{x_v}$ after an aggregation is positively correlated with the selection of Client v.*

Proof. Based on Equation (6), the global weight w^t in the tth round of aggregation is calculated as:

$$\begin{aligned} w^t &= \frac{1}{\sum_i \xi_i} \sum_i \xi_i w_i^t \\ &= w^{t-1} - \frac{\eta}{M} \sum_i \xi_i \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_i})}{\partial w_i^{t-1}}, \end{aligned} \tag{7}$$

where $\xi_i \in \{0, 1\}$ denotes whether Client i is selected for the tth round of aggregation, $M = \sum_i \xi_i$ is the number of selected clients. Here we use ξ_i instead of ξ_i^t for simplicity.

We denote $\text{Exposure}_{w^t}(\boldsymbol{x_v})$ as the exposure rate of the victim Client v's private data $\boldsymbol{x_v}$ at round t. Then the change of the exposure rate after an aggregation can be calculated as:

$$\begin{aligned} &\text{Exposure}_{w^t}(\boldsymbol{x_v}) - \text{Exposure}_{w^{t-1}}(\boldsymbol{x_v}) \\ =& -[\mathcal{L}(w^t, \boldsymbol{x_v}) - \mathcal{L}(w^{t-1}, \boldsymbol{x_v})] \\ =& -[\mathcal{L}(w^{t-1} - \frac{\eta}{M} \sum_i \xi_i \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_i})}{\partial w^{t-1}}, \boldsymbol{x_v}) - \mathcal{L}(w^{t-1}, \boldsymbol{x_v})] \\ \approx& \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_v})}{\partial w^{t-1}} \frac{\eta}{M} \sum_i \xi_i \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_i})}{\partial w^{t-1}} \\ &(\text{we define} \quad g(\boldsymbol{x_i}) \triangleq \frac{\partial \mathcal{L}(w^{t-1}, \boldsymbol{x_i})}{\partial w^{t-1}}) \\ =& \frac{\eta}{M} g(\boldsymbol{x_v}) \sum_i \xi_i g(\boldsymbol{x_i}) = \frac{\eta}{M} \left(\xi_v g^2(\boldsymbol{x_v}) + g(\boldsymbol{x_v}) \sum_{i \neq v} \xi_i g(\boldsymbol{x_i}) \right). \end{aligned} \tag{8}$$

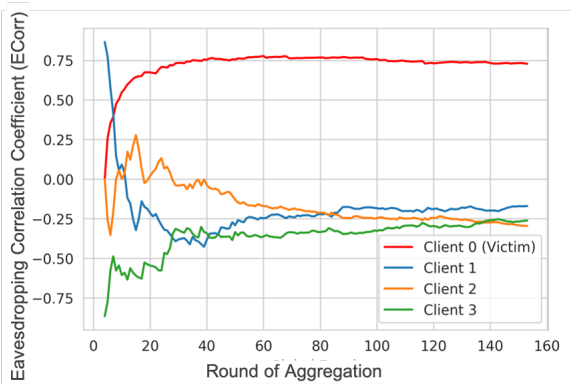


Fig. 2: Eavesdropping Correlation Coefficients during FL. Four clients collaboratively train a global model. To reveal the client identity of the record of interest (a private record belonging to Client 0), the adversary calculates the ECorr between the selection of clients and the exposure rates of the record of interest. The victim client (Client 0 in red) presents the highest ECorr among the four clients. Hence, the adversary can pinpoint the victim client by observing the highest ECorr. Because $\frac{\eta}{M} g^2(\boldsymbol{x_v}) \geq 0$, the change of exposure rates of $\boldsymbol{x_v}$ is positively correlated with the selection ξ_v. □

B. Eavesdropping Attack

In Eavesdropping attack, we assume that for every aggregation in FL, the adversary snapshots weights of the global model and knows whether the victim client is selected in the aggregation. For example, the adversary monitors the communication around the victim client to acquire information about the client selection. Based on Theorem 1, if a client is selected for aggregation, then the training data of this participant will be more exposed after aggregation than that of other unselected participants. Therefore, we can derive the client's private record by selecting the exposed data with the highest correlation coefficient. According to Equation (8), the correlation relationship between the change of exposure rates and client selection is non-linear. We leverage Spearman's rank correlation coefficient [33] to assess the non-linear relationship using rank order. Given a sequential text $\boldsymbol{x}$, we derive Eavesdropping Correlation Coefficient ECorr as:

$$\begin{aligned} \text{ECorr} \triangleq \quad & \text{Spearman}\Big(\{\text{Exposure}_{\mathcal{F}_w^t}(\boldsymbol{x}) \\ & - \text{Exposure}_{\mathcal{F}_w^{t-1}}(\boldsymbol{x})\}, \{s^t\} \Big), \end{aligned} \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089713, IEEE Internet of Things Journal

Algorithm 1 Eavesdropping Attack

INPUT: Number of aggregation rounds T, the index of victim client i, the global model $\mathcal{F}_w^t$, and the selection indicator of victim client s_t at the tth round of aggregation.
OUTPUT: Prediction of private record $\boldsymbol{x}^*$.

1: **for** $t \leftarrow 1$ to T **do**
2: Record the selection of the victim client s_t.
3: Snapshot the global model $\mathcal{F}_w^t$.
4: **end for**
5: Select candidate records $\hat{\mathcal{X}}$.
6: Calculate Eavesdropping Correlation Coefficient ECorr for all $\boldsymbol{x} \in \hat{\mathcal{X}}$.
7: Update rank $r(\boldsymbol{x})$ for all $\boldsymbol{x} \in \hat{\mathcal{X}}$.
8: Output the record with the highest rank, $\boldsymbol{x}^* \leftarrow \underset{\boldsymbol{x}\in\hat{\mathcal{X}}}{\text{argmin}}\ r(\boldsymbol{x})$.

where Spearman$(\cdot,\cdot)$ calculates the Spearman's rank correlation coefficient, and s^t denotes an indicator function that outputs 1 if the victim client is selected in aggregation at the tth aggregation round, and -1 otherwise. Here we follow the common notation in Spearman's rank correlation and let $s^t = 2\xi^t - 1$. Eavesdropping Correlation Coefficient measures how likely the exposure of a record correlates with a client. Figure 2 illustrates the Eavesdropping Correlation Coefficients during FL. The adversary can reveal the victim client's identity by pinpointing the client with the highest ECorr. Therefore, by calculating the correlation between exposure rates based on snapshotted global models and clients' selections, the adversary can match the client identity with the records and then expose both client identity and the records of interest.

Algorithm 1 summarizes Eavesdropping attack. In Eavesdropping attack, the adversary records the selection of the victim client for every aggregation and snapshots the updated FL models. After federated training, the adversary analyzes the correlation.

In the early training phase, many records close to the private records (*e.g.*, small edit distances) may achieve similar or even higher coefficient rates than the private records. These data samples may mislead the correlation measurement. To eliminate the misleading correlation, we derive the correlation rank by combining the rank of Eavesdropping Correlation Coefficient and the rank of the final exposure rates.

$$r(\boldsymbol{x}) = \text{rank}\Big(\text{ECorr}(\boldsymbol{x})\Big) + \text{rank}\Big(\text{Exposure}_{\mathcal{F}_w^T}(\boldsymbol{x})\Big), \qquad (10)$$

where rank$(\cdot)$ calculates the rank (ascending order) of a given data in the candidate list, and T denotes the last round of aggregation. Finally, the adversary outputs the record x^* with the highest rank $r(x^*)$ as the private record.

C. Watermark Attack

If the adversary cannot access the selection of clients, Watermark attack brings a new approach to generate correlation. In Watermark attack, we assume that the adversary can inject a record (watermark) into a victim client's dataset, but cannot read the data. For example, many individuals (*e.g.*, doctors, pharmacists, therapists) can contribute to a hospital's database (*i.e.*, local data of a client), but cannot read the whole database. An insider attacker in the hospital (*e.g.*, a compromised device of a doctor) may inject a watermark into the hospital's database so as to extract other individuals' data. We hypothesize that the exposure rates of records from the same client change similarly after each aggregation. Therefore, by comparing the correlation of the changes of exposure rates between the watermark and the potential records, the adversary extracts the records of interest. Given two sequential text $\boldsymbol{x}$ and $\boldsymbol{y}$, we derive Watermark Correlation Coefficient WCorr as follows.

$$\text{WCorr} \triangleq \text{Spearman}\Big(\{\text{Exposure}_{\mathcal{F}_w^t}(\boldsymbol{x}) - \text{Exposure}_{\mathcal{F}_w^{t-1}}(\boldsymbol{x})\}, \{\text{Exposure}_{\mathcal{F}_w^t}(\boldsymbol{y}) - \text{Exposure}_{\mathcal{F}_w^{t-1}}(\boldsymbol{y})\}\Big). \qquad (11)$$

We input the candidate records as $\boldsymbol{x}$ and the watermark as $\boldsymbol{y}$. The higher correlation coefficient indicates the higher probability that the record of interest belongs to the victim client. Figure 3 illustrates the Watermark Correlation Coefficients during FL. The adversary can pinpoint the victim client's identity by observing the client with the highest WCorr.

Algorithm 2 summarizes Watermark attack. First, the adversary randomly generates a record and injects it into the victim client. We refer to the record as a watermark, which indicates whether the record is involved in the model aggregation or, in other words, whether the victim client is selected for aggregation. The watermark helps differentiate the victim client from others. Note that most data injection attacks (*e.g.*, data poisoning, backdoor attacks) investigate the robustness (integrity) of machine learning models. This is the first attempt to explore the privacy leakage caused by data injection. To avoid detection, the adversary randomly generates the watermark without a fixed pattern and injects the watermark only once. The adversary snapshots the global models in each aggregation and measures the correlation between the exposure of the watermark and that of the records of interest.

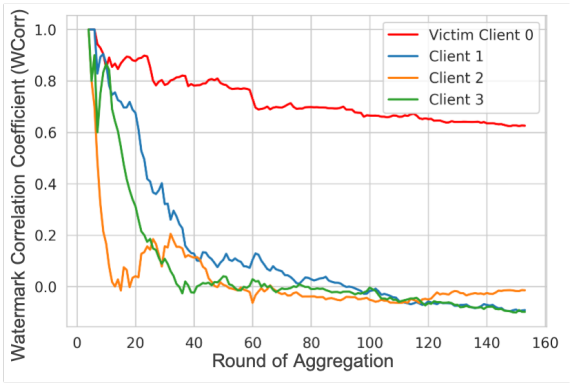


Fig. 3: Watermark Correlation Coefficients during FL. Four clients collaboratively train an FL model. To reveal the client identity of the record of interest (a private record belonging to Client 0), the adversary injects a watermark into Client 0's data and calculates the WCorr between the exposure rates of the record of interest and those of the watermark. The victim client (Client 0 in red) presents the highest WCorr among the four clients.

Algorithm 2 Watermark Attack

INPUT: Number of aggregation rounds T, the index of victim client j, the global model $\mathcal{F}_w^t$ at round t.
OUTPUT: Prediction of private record $\boldsymbol{x}^*$.

1: Randomly generate a watermark y.
2: Injects a watermark y into the client j's training data.
3: **for** $t \leftarrow 1$ to T **do**
4: Snapshot the global model $\mathcal{F}_w^t$.
5: **end for**
6: Select candidate records $\hat{\mathcal{X}}$.
7: Calculate Watermark Correlation Coefficient WCorr for all $\boldsymbol{x} \in \hat{\mathcal{X}}$.
8: Update rank $r(\boldsymbol{x})$ for all $\boldsymbol{x} \in \hat{\mathcal{X}}$.
9: Output the record with the highest rank, $\boldsymbol{x}^* \leftarrow \underset{\boldsymbol{x} \in \hat{\mathcal{X}}}{\text{argmin}} \; r(\boldsymbol{x})$.

Following the same idea as in Eavesdropping attack, we mitigate the extensive computation and misleading correlation. Thus, we derive a correlation ranking r based on the exposure rates of the final round and the Watermark Correlation Coefficients:

$$r(\boldsymbol{x}) = \text{rank}\Big(\text{WCorr}(\boldsymbol{x})\Big) + \text{rank}\Big(\text{Exposure}_{\mathcal{F}_w^T}(\boldsymbol{x})\Big). \quad (12)$$

In this way, the adversary in Watermark attack predicts the record x^* with the highest rank $r(x^*)$ as the private record.

V. Experimental Evaluation

In this section, we apply our proposed correlation attacks to a language modeling task, which is one of the most widely used FL applications [8]. We empirically evaluate our methodology on several popular model architectures and standard datasets.

A. Datasets, Model Architectures, and Training Setup

We evaluate the attacks on three datasets: Penn Treebank (PTB) [34], WikiText-2 (Wiki2) [35], and Enwik8 [36]. The PTB dataset is a small dataset containing about 5MB text from newspapers. The WikiText-2 dataset consists of about 11MB text extracted from Wikipedia. The Enwik8 dataset is the largest one containing about 100MB English Wikipedia text. We lowercase all the characters in the datasets and separate all datasets into three parts: training, validation, and testing. In addition, we filter the Enwik8 dataset to clean texts, which contains only 26 letters a through z, 10 digits 0 through 9, and spaces.

We train character-level language models using a three-layer LSTM model [27]. We deploy an embedding layer with 128 units and three LSTM layers with 128 units each for the PTB dataset, 512 units for the Wiki2 dataset, and 1024 units for the Enwik8 dataset. In the experiment, four clients jointly train an FL model, and in each aggregation round, two clients will be selected to upload their models for aggregation (We will evaluate the impact of different numbers of clients In Section V-C). Each client uses 1/4 of the training dataset without overlapping as their training data. We randomly sample social security numbers from a uniform distribution and add a private record "my social security number is xxx-xx-xxxx" to each client's training dataset. We insert the private record four times (eight times for the Enwik8 dataset) into each client's training dataset. We will discuss the impact of the number of record insertions in Section V-C. If not explicitly mentioned, we use the above setting as default in the experiments.

We adopt Adam [37] as the optimizer with an initial learning rate of 0.001 and batch size 128 to train the models and reduce the learning rate by 10 if the validation loss does not decrease in the last five rounds. We clip gradients in training to avoid exploding gradients (gradients are clipped if greater than 1 or less than -1). Clients train the models for two (local) epochs between each asynchronous aggregation. In total, the FL model is aggregated for 400 rounds. In addition, overfitting is one of the reasons that neural networks can remember the input data. Therefore, to avoid overfitting, we carefully design our training process using dropout and early stopping: 1) We deploy dropout in the models. 10% dropout is applied to the output of the LSTM layers to avoid overfitting; 2) The server monitors the performance of the models on the validation datasets and stops the training if the validation loss does not decrease for several rounds of aggregation.

B. Performance Evaluation

Evaluation Metrics. We evaluate these two attacks using the following two metrics. 1) Top-K accuracy: we calculate the probability that whether the private record of the victim client matches the top-K record candidates inferred by the adversary. The correct inference indicates that the adversary can predict both the content and the client identity of the private record correctly. 2) Top-K smallest edit distance: we calculate the smallest edit distance of the distances between top-K record candidates and the private record. This indicates how close the predicted record to the private record.

We report Bit per Character (BPC) to measure the average performance of language models for characters over the test dataset. Given a text $x_1 \cdots x_n$ and a language model $\mathcal{F}_w$ with parameters w, BPC is defined as $\log_2$ Perplexity.

Baseline Attack. Carlini *et al*. showed that their approach could expose the sensitive text record in standalone machine learning [20]. We use their approach as our baseline and compare it with our proposed Eavesdropping Attack and Watermark Attack.

We report the accuracy and smallest distances achieved by the three approaches (Baseline Attack, Eavesdropping Attack, and Watermark Attack) on the PTB, Wiki2, and Enwik8 dataset (Table I, II, III, IV, V, and VI). Top-1 accuracy shows that the success rate of adversaries predicting the confidential records of clients with one guess. From the results, we observe much higher attack accuracies achieved by our proposed Eavesdropping Attack and Watermark Attack, compared with Baseline Attack. When the adversaries have multiple chances to guess the confidential records (*e.g*., 50 guesses), we observe that the adversaries can achieve no less than 80% accuracies (Top-50 accuracy in the tables), which poses severe privacy threats to the clients. For example, to extract records from

TABLE I: Accuracy comparison on the PTB dataset.

Accuracy	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	0.25	0.98	0.80
Top-5	0.75	0.98	0.80
Top-10	0.83	0.98	0.80
Top-20	0.85	1.00	0.90
Top-50	0.95	1.00	1.00

TABLE II: The smallest edit distance comparison on the PTB dataset.

Distance	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	5.85	0.08	1.40
Top-5	1.73	0.05	1.40
Top-10	1.08	0.05	1.40
Top-20	0.88	0.00	0.30
Top-50	0.20	0.00	0.00

TABLE III: Accuracy comparison on the Wiki2 dataset.

Accuracy	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	0.25	0.58	0.30
Top-5	0.38	0.73	0.50
Top-10	0.45	0.75	0.60
Top-20	0.53	0.75	0.60
Top-50	0.55	0.80	0.70

TABLE IV: The smallest edit distance comparison on the Wiki2 dataset.

Distance	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	5.78	1.65	2.10
Top-5	4.03	1.20	1.70
Top-10	3.48	1.10	1.00
Top-20	2.98	1.03	1.00
Top-50	2.58	0.90	0.60

TABLE V: Accuracy comparison on the Enwik8 dataset.

Accuracy	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	0.25	0.50	0.10
Top-5	0.50	0.68	0.20
Top-10	0.55	0.78	0.30
Top-20	0.63	0.78	0.40
Top-50	0.63	0.80	0.80

the PTB dataset, our proposed attacks can achieve 100% accuracies after 20 guesses (Top-20 accuracy). Furthermore, according to the results of edit distances, the extracted records derived from our proposed attacks are very close to the confidential records of clients (no more than one character different after 50 guesses). With the increase of clients' data sizes (from PTB to WIki2 to Enwik8), the performance of record extraction will be slightly decreased, but it is still highly likely to leak the record information based on the three evaluated datasets. *The experimental results demonstrate that both Eavesdropping Attack and Watermark Attack can extract exact records and reveal their client identities with much higher accuracy rates and smaller edit distances compared to Baseline Attack.*

TABLE VI: The smallest edit distance comparison on the Enwik8 dataset.

Distance	Baseline attack	Eavesdropping attack	Watermark attack
Top-1	5.63	1.53	3.30
Top-5	3.33	1.08	2.60
Top-10	2.80	0.73	2.30
Top-20	2.43	0.70	1.90
Top-50	2.23	0.63	1.00

C. Ablation Study

We now investigate the performance of attacks under different settings: different numbers of private record insertions $(1, 2, 4, 8)$, different numbers of local epochs $(1, 2, 4, 8)$, and different numbers of clients $(4, 8, 16, 32)$. Figure 4 and 5 demonstrate the (Top-1) accuracy and edit distance under different settings.

We first investigate the impact of the number of private records in the victim client's data. Hence, we insert the records 1, 2, 4, and 8 times to the client's local data. Record-level privacy is more likely to be exposed with more record insertions in clients' data (Figure 4a and 5a). We observe that the increasing number of local epochs does not have a significant impact on the proposed attacks (Figure 4b and 5b). However, with an increasing number of clients, the risk of Eavesdropping Attack and Watermark Attack will be reduced (Figure 4c and 5c). Since, in our experimental setting, we assume each client has a confidential record with a similar pattern (starting with "my social security number is " and followed by nine digits), with the increasing number of clients, the similar records learned by the global model increase the difficulties of adversaries to differentiate the specific record from the victim client. That is to say, if not many clients ($N \leq 16$) are involved in the FL, it is highly likely that the private records and client identities will be exposed. Comparing Eavesdropping Attack and Watermark Attack in all the settings, we find that Eavesdropping Attack causes higher risk in most of the settings, but the risk gap can be narrowed down with the increasing number of record insertions and local epochs.

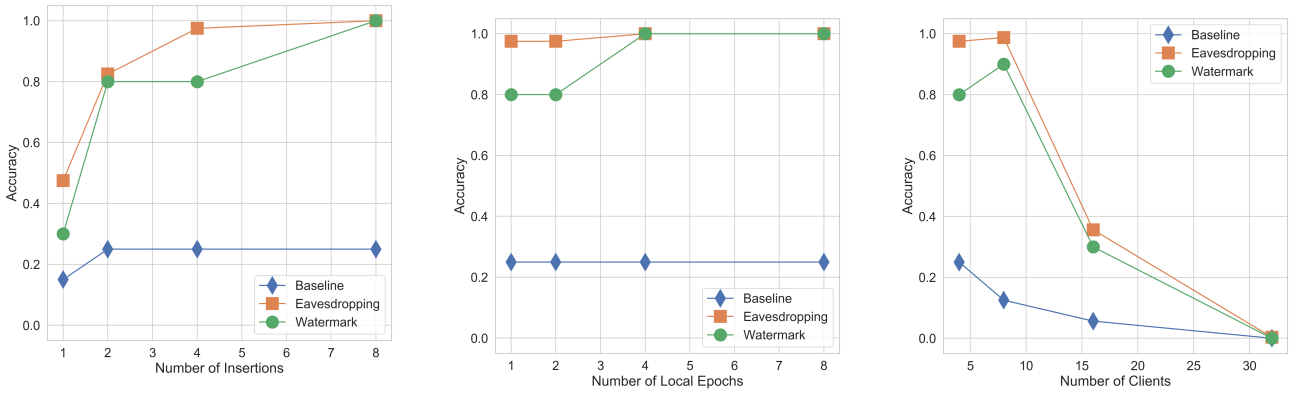
VI. Countermeasures

In this section, we investigate three potential countermeasures to mitigate the record-level privacy leakage due to Eavesdropping attack and Watermark attack.

A. Obfuscate Word Embedding

Word embedding is a widely-used neural network module for almost all NLP tasks, which encodes discrete text inputs to a continuous embedding space. However, word embedding is likely to leak sensitive data information - if a record is used for training, its corresponding parameters in the embedding layer are more likely to be changed. We propose three approaches to obfuscate the information collected in word embedding during training and prevent record-level privacy leakage, including Noised Embedding, Dropout Embedding, and Adversarial Embedding, which are given below, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089713, IEEE Internet of Things Journal

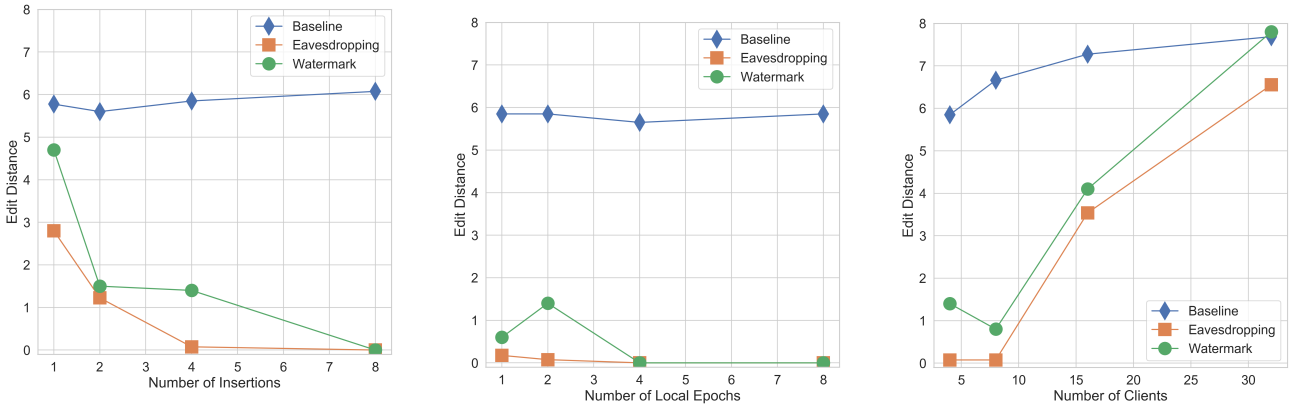


(a) Different number of private record insertions.

(b) Different number of local epochs.

(c) Different number of clients.

Fig. 4: Comparison of accuracy performance under different settings: different number of private record insertions, different number of local epochs, and different number of clients.



(a) Different number of private record insertions.

(b) Different number of local epochs.

(c) Different number of clients.

Fig. 5: Comparison of edit distance performance under different settings: different number of private record insertions, different number of local epochs, and different number of clients.

- *Noised Embedding*. We add noises to the output of the embedding layer during clients' local training. The noises are sampled from a Gaussian distribution $\mathcal{N}(0, \sigma)$. We report the results with different noise scales ($\sigma \in \{0.01, 0.001, 0.0001\}$).
- *Dropout Embedding*. We randomly dropout part of the embedding outputs during clients' local training to mitigate the record leakage. We report the results with different dropout rates ($r \in \{0\%, 10\%, 20\%, 30\%, 40\%, 50\%\}$).
- *Adversarial Embedding*. Adversarial examples are a small perturbation to the input samples that mislead the outputs of a well-trained model in the inference phase. Adversarial examples degrade the robustness of a well-trained model. Next, we investigate the relationship between model robustness and privacy preservation. We increase the model robustness by adversarial training - introducing adversarial examples in the training phase for every client. We follow the virtual adversarial training proposed in [38]. Specifically, we generate adversarial perturbations to the embedding layer and train the language model with both the original embedding vectors and the adversarial embedding vectors. The perturbation added to the embedding layer can be calculated as:

$$\delta = -\epsilon \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}, \tag{13}$$

where $\boldsymbol{g}$ is the gradient of original word embedding vectors derived from the loss function. We compare different magnitudes on the perturbations ($\epsilon \in \{1, 0.1, 0.01\}$).

TABLE VII: Impact of Noised Embedding on Eavesdropping attack and Watermark attack.

	Noise scale (σ)	Accuracy	Distance	BPC
Eavesdropping attack	0.01	0.00	7.50	3.62
	0.001	0.75	0.30	1.38
	0.0001	0.80	0.25	1.40
Watermark attack	0.01	0.00	7.70	3.62
	0.001	0.80	0.60	1.38
	0.0001	0.80	0.80	1.39

In the experiment, we find that Noised Embedding can reduce the attack accuracy to 0%, but, in the meantime, degrade the model performance to a low BPC of 3.62 (Table VII). Adversarial Embedding does not help reduce the privacy risk against Eavesdropping attack (Table IX). Adversarial Embed-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3089713, IEEE Internet of Things Journal

TABLE VIII: Impact of Dropout Embedding on Eavesdropping attack and Watermark attack.

	Dropout rate (r)	Accuracy	Distance	BPC
Eavesdropping attack	0%	0.98	0.08	1.44
	10%	0.90	0.40	1.38
	20%	0.93	0.45	1.37
	30%	0.73	0.75	1.38
	40%	0.68	1.05	1.38
	50%	0.65	0.95	1.38
Watermark attack	0%	0.50	1.80	1.43
	10%	0.80	0.20	1.38
	20%	0.90	0.10	1.36
	30%	0.70	0.30	1.36
	40%	0.70	0.30	1.36
	50%	0.50	0.90	1.37

TABLE IX: Impact of Adversarial Embedding on Eavesdropping attack and Watermark attack.

	Adversarial magnitude (λ)	Accuracy	Distance	BPC
Eavesdropping attack	1.00	0.88	0.38	1.41
	0.10	1.00	0.00	1.38
	0.01	1.00	0.00	1.39
Watermark attack	1.00	0.90	0.10	1.37
	0.10	0.80	0.40	1.39
	0.01	1.00	0.0	1.38

TABLE X: Impact of LSTM Dropout on Eavesdropping attack and Watermark attack.

	Dropout rate (r)	Accuracy	Distance	BPC
Eavesdropping attack	0%	0.98	0.20	1.43
	10%	0.98	0.08	1.44
	20%	1.00	0.00	1.36
	30%	0.93	0.10	1.38
	40%	0.23	3.05	1.44
	50%	0.05	4.95	1.46
Watermark attack	0%	0.50	1.80	1.43
	10%	0.80	0.20	1.37
	20%	0.40	2.10	1.41
	30%	0.50	2.40	1.40
	40%	0.20	4.00	1.44
	50%	0.00	5.50	1.46

TABLE XI: Impact of Differential Privacy on Eavesdropping attack and Watermark attack.

	Noise scale (σ)	Accuracy	Distance	BPC
Eavesdropping attack	0.01	0.98	0.10	1.39
	0.10	0.03	5.05	1.51
	1.00	0.00	7.48	1.96
Watermark attack	0.01	1.00	0.00	1.37
	0.10	0.60	1.70	1.44
	1.00	0.00	7.70	1.78

ding even increases the privacy risk against Watermark attack, which was also observed in [39]. *Dropout Embedding alone does mitigate the risk of record leakage while maintaining model performance (Table VIII), but the proposed attacks could still achieve 50% accuracy (success rate) in record extraction.*

B. Avoid Overfitting

Overfitting is considered as one of the potential reasons for privacy leakage: the language model performs perfectly on the training data, but cannot be generalized to the test data. We apply dropout to the outputs of LSTM layers to prevent overfitting. To avoid ambiguity with Dropout Embedding, we name this approach LSTM Dropout. We vary the dropout rates r from 0% to 50% and investigate their impact on privacy leakage. We report the Top-1 accuracy and the smallest distance in Table X. We observe that dropout does not significantly reduce the privacy risk caused by Eavesdropping attack and Watermark attack. Higher dropout rates may prevent privacy leakage, but lower the models' performance as well. The defender may need to balance the risk of privacy leakage and model performance.

C. Differential Privacy

Differential privacy is a strategy to bound the individual information exposure when running an algorithm f. In this paper, we use (ϵ, δ)-differential privacy to measure the privacy of the FL model. An algorithm f is (ϵ, δ)-differential private on a dataset $\mathcal{D}$ if

$$\Pr(f(\mathcal{D}) \in S) \leq \delta + e^{\epsilon} \Pr(f(\mathcal{D}') \in S), \tag{14}$$

for any set S of possible outputs of f and neighbor dataset D'. We apply the differentially private SGD algorithm (DP-SGD) [40] to mitigating the privacy leakage. DP-SGD clips the models' gradient and adds Gaussian noise to the gradients to fulfill (ϵ, δ)-differential privacy. We investigate three different privacy budgets (ϵ, δ) by adding Gaussian noise $\mathcal{N}(0, \sigma)$. We evaluate our approaches with different noise scales ($\sigma \in \{0.01, 0.1, 1\}$).

From the experimental results (Table XI), we observe that differential privacy can reduce the impact of Eavesdropping attack and Watermark attack, but at the cost of degrading the models' performance. With a larger noise scale σ, differential privacy may reduce the data exposure to 0% accuracy, while the performance of the model is reduced to 1.77 BPC. Therefore, the utility-privacy trade-off is critical for developing FL models, which was also observed in standalone models [41].

D. Secure Aggregation

Secure Aggregation, as one of the Secure Multiparty Computation (SMC) algorithms, has been applied to protect the privacy of clients' model [5], [6], [18], [19]. Clients encrypt their private models through secure aggregation algorithms, and the server can only decrypt the sums of the model parameters. Clients' local models are not exposed to the server and other clients using secure aggregation. However, our proposed approaches do not require the knowledge of clients' local models. Therefore, encrypting local models using secure aggregation is not applicable to reduce the risk of record-level privacy.

In summary, among all the countermeasures we investigate, most approaches need significantly sacrifice the model performance to reduce privacy risk (*e.g.*, Noise Embedding, LSTM

Dropout, and Differential Privacy). Adversarial Embedding and Secure Aggregation are limited in preventing record-level privacy leakage. Only Dropout Embedding can partially mitigate the risk of record-level privacy while maintaining the model performance. Unfortunately, the identified attacks in this paper could still achieve 50% accuracy in record extraction when applying Dropout Embedding as a defense.

VII. Conclusion

In this paper, we have made the first attempt to explore the record-level data leakage in federated learning even without access to the victim client's local model. We have developed a unified framework, under which two correlation attacks, Eavesdropping attack and Watermark attack, could extract clients' private records and reveal client identities. Through extensive studies, we have demonstrated the effectiveness of the identified attacks on three widely used language modeling datasets. Accordingly, we have investigated several countermeasures against such attacks. We have discovered that most countermeasures do reduce the risk of record extraction but significantly sacrifice model performance, and only applying dropout in word embedding could mitigate the risk of record leakage while maintaining model performance. Unfortunately, the identified attacks could still achieve 50% accuracy in record extraction. This paper has indeed demonstrated the real privacy threats for the existing federated learning. We hope that our work serves as the first step to facilitate the future privacy research on privacy protection in emerging future IoT systems.

Acknowledgment

This work was supported in part by National Science Foundation (CNS-1747783, CCF-2007210) and Industrial Members of NSF Center for Big Learning (CBL). The work of Zhang and Fang was supported in part by US National Science Foundation under IIS-1722791.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[3] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konecny, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," in *Proceedings of the 2nd SysML Conference, Palo Alto*, 2019.

[4] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "Control of photo sharing over online social networks," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 704–709.

[5] K. Xu, H. Ding, L. Guo, and Y. Fang, "A secure collaborative machine learning framework based on data locality," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–5.

[6] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "My privacy my decision: Control of photo sharing on online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 2, pp. 199–210, 2015.

[7] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[8] "Federated learning: Collaborative machine learning without centralized training data," Apr 2017. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[9] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[10] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "Ffd: A federated learning based method for credit card fraud detection," in *International Conference on Big Data*. Springer, 2019, pp. 18–32.

[11] U. M. Aïvodji, S. Gambs, and A. Martin, "Iotfla: A secured and privacy-preserving smart home architecture implementing federated learning," in *2019 IEEE Security and Privacy Workshops*. IEEE, 2019, pp. 175–180.

[12] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," in *IEEE Symposium on Security and Privacy*, 2019.

[13] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy*. IEEE, 2019, pp. 691–706.

[14] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the gan: information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 603–618.

[15] W. Wei, L. Liu, M. Loper, K.-H. Chow, M. E. Gursoy, S. Truex, and Y. Wu, "A framework for evaluating gradient leakage attacks in federated learning," *arXiv preprint arXiv:2004.10397*, 2020.

[16] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2019, pp. 2512–2520.

[17] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients–how easy is it to break privacy in federated learning?" *arXiv preprint arXiv:2003.14053*, 2020.

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[19] Y. Gong, Y. Fang, and Y. Guo, "Private data analytics on biomedical sensing data via distributed computation," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 13, no. 3, pp. 431–444, 2016.

[20] N. Carlini, C. Liu, Ú. Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," in *28th USENIX Security Symposium*, 2019, pp. 267–284.

[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[22] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *International Conference on Learning Representations (ICLR)*, 2020.

[23] B. Krause, E. Kahembwe, I. Murray, and S. Renals, "Dynamic evaluation of neural sequence models," in *International Conference on Machine Learning (ICML)*, 2018, pp. 2766–2775.

[24] G. Melis, C. Dyer, and P. Blunsom, "On the state of the art of evaluation in neural language models," *International Conference on Learning Representations (ICLR)*, 2018.

[25] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing lstm language models," *International Conference on Learning Representations (ICLR)*, 2018.

[26] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *International Conference on Learning Representations (ICLR)*, 2017.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

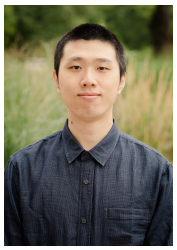
[28] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber, "Recurrent highway networks," in *International Conference on Machine Learning (ICML)*, 2017.

[29] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[30] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 774–14 784.

[31] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *arXiv preprint arXiv:2001.02610*, 2020.

[32] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. C. Lai, and R. L. Mercer, "An estimate of an upper bound for the entropy of english," *Computational Linguistics*, vol. 18, no. 1, pp. 31–40, 1992.
[33] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 100, no. 3/4, pp. 441–471, 1987.
[34] M. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: The penn treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.
[35] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *International Conference on Learning Representations (ICLR)*, 2017.
[36] M. Mahoney, "Large text compression benchmark."
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.
[38] T. Miyato, A. M. Dai, and I. Goodfellow, "Adversarial training methods for semi-supervised text classification," *International Conference on Learning Representations (ICLR)*, 2017.
[39] L. Song, R. Shokri, and P. Mittal, "Privacy risks of securing machine learning models against adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, p. 241–257.
[40] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.
[41] B. Jayaraman and D. Evans, "Evaluating differentially private machine learning in practice," in *28th USENIX Security Symposium*, 2019, pp. 1895–1912.

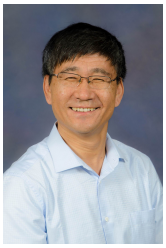


Xiaoyong Yuan received the B.S. degree from Fudan University, in 2012, the M.E. degree from Peking University, in 2015, and the Ph.D degree from University of Florida, in 2020.

Dr. Yuan is an assistant professor at the College of Computing, Michigan Technological University, Houghton, MI. His research interest mainly spans the fields of machine learning, deep learning, security and privacy, and cloud computing.

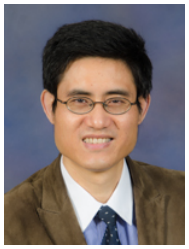
Xiyao Ma is a PhD student in NSF Center for Big Learning, Department of Electrical and Computer Engineering, University of Florida. His research interests include Natural Language Generation and Understanding, Graph Neural Networks, and Reinforcement Learning.

Lan Zhang received the B.S. degree and M.S. degree from the University of Electronic Science and Technology of China in 2013 and 2016, respectively, and the Ph.D. degree from the University of Florida in 2020. Dr. Zhang is an assistant professor in the Department of Electrical and Computer Engineering at Michigan Technological University, Houghton, MI. Her research interest mainly spans wireless communications and distributed machine learning in various cyber-physical systems and Internet of Things applications.



Yuguang Fang (F'08) received an MS degree from Qufu Normal University, Shandong, China in 1987, a PhD degree from Case Western Reserve University in 1994, and a PhD degree from Boston University in 1997. He joined the Department of Electrical and Computer Engineering at University of Florida in 2000 as an assistant professor, then received early promotion to associate professor with tenure in 2003 and full professor in 2005, and has been a distinguished professor since 2019. He holds a University of Florida Foundation Preeminence Term Professorship (2019-2022), a University of Florida Research Foundation Professorship (2017-2020, 2006-2009), a University of Florida Term Professorship (2017-2019, 2019-2021).

Dr. Fang received the US NSF Career Award in 2001, the US ONR Young Investigator Award in 2002, the 2018 IEEE Vehicular Technology Outstanding Service Award, 2019 IEEE Communications Society AHSN Technical Achievement Award, the 2015 IEEE Communications Society CISTC Technical Recognition Award, the 2014 IEEE Communications Society WTC Recognition Award, the Best Paper Award from IEEE ICNP (2006), and the 2010-2011 UF Doctoral Dissertation Advisor/Mentoring Award. He was the Editor-in-Chief of IEEE Transactions on Vehicular Technology (2013-2017) and IEEE Wireless Communications (2009-2012), and serves/served on several editorial boards of premier journals. He also served as the Technical Program Co-Chair of IEEE INFOCOM'2014. He is a fellow of IEEE and AAAS.



Dapeng Wu (S'98–M'04–SM'06–F'13) received a B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 1990, an M.E. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1997, and a Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003.

He is a professor at the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security. He received University of Florida Term Professorship Award in 2017, University of Florida Research Foundation Professorship Award in 2009, AFOSR Young Investigator Program (YIP) Award in 2009, ONR Young Investigator Program (YIP) Award in 2008, NSF CAREER award in 2007, the IEEE Circuits and Systems for Video Technology (CSVT) Transactions Best Paper Award for Year 2001, and the Best Paper Awards in IEEE GLOBECOM 2011 and International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine) 2006.