Information Leaks in Federated Learning

Anastasia Pustozerova* and Rudolf Mayer*
{apustozerova,rmayer}@sba-research.org
*SBA Research, Vienna, Austria

***Abstract*—With the surge in data collection and analytics, concerns are raised with regards to the privacy of the individuals represented by the data. In settings where the data is distributed over several data holders, federated learning offers an alternative to learn from the data without the need to centralize it in the first place. This is achieved by exchanging only model parameters learned locally at each data holder. This greatly limits the amount of data to be transferred, reduces the impact of data breaches, and helps to preserve the individual's privacy. Federated learning thus becomes a viable alternative in IoT and Edge Computing settings, especially if the data collected is sensitive.**
However, risks for data or information leaks still persist, if information can be inferred from the models exchanged. This can e.g. be in the form of membership inference attacks. In this paper, we investigate how successful such attacks are in the setting of sequential federated learning. The cyclic nature of model learning and exchange might enable attackers with more information to observe the dynamics of the learning process, and thus perform a more powerful attack.

I. Introduction

Machine learning (ML) is widely used for data analysis and demands a vast amount of data for better predictions. The data used for training these models often has private nature or is otherwise sensitive, e.g. businesses relevant, and owners of the data may be uncomfortable with sharing it. They may use techniques for sanitizing data before sharing, to avoid possible risks of inference. For example, k-anonymity is a method for preserving the anonymity of individuals, by modifying the attributes of the dataset in a way that each instance has at least $k-1$ other entities with identical *quasi-identifiers* [1]. The method is sensitive to outliers and was shown to be vulnerable to several types of inference attacks [2]. Differential privacy [3] allows to hide individuals' information and at the same time preserves statistical properties of the dataset. In settings where the same or similar analysis needs to be repeated several times, the approach has limitations, as the privacy budget may be used up quickly. Further, if the interest of the attacker is on global properties, these methods provide only limited protection.

Unlike traditional learning, when data needs to be centralized for processing, *federated learning* addresses the problem of data ownership and locality. It allows training machine learning models locally at each user's site, thus avoiding data centralization, and communicating only model parameters instead. IoT is a candidate for applying federated learning, due to its distributed nature, its decentralized data recording, and potentially very sensitive data that some devices might capture. Algorithms improving and adapting federated learning for IoT specifications were recently proposed [4]–[6].

Recent studies [7]–[9] show that certain ML models can still leak information about the data on which they were trained. Therefore, these models become the target for adversaries. Federated learning might even provide a larger attack surface. In this work, we evaluate the risks of information leakage from neural network models by performing membership inference attacks by an insider, in a sequential federated learning setting, where models are trained in cycles. We show that this setting is more vulnerable to membership inference attacks by an insider than in centralized settings. The insider is assumed to be of the *honest but curious* adversary model, i.e. a legitimate participant in the learning that will not deviate from the protocol, but will attempt to infer information from legitimately received messages.

We note that a higher risk of membership inference occurs if two or more malicious parties of the federated learning cooperate and exchange the observed models, which then reflect different stages of training. This enables intruders to see the dynamics of the learning, as its performance changes after each training at the following party. We show that the number of nodes in federated learning, as well as the number of data points at each node, influence the accuracy of the attack. We further propose different mitigation strategies that can be applied in federated learning settings to reduce the effectiveness of membership inference attacks.

II. Related Work

Federated Learning is a method to leave data distributed at the site where it originates, e.g. on mobile devices, or where it is initially gathered, but to still learn a global model, based on parameters aggregated from local training [10]. The idea of local training is relevant in particular for personal data, where data sharing brings regulatory, privacy and technical issues. Federated learning approaches were thus applied on various tasks in medical domain [11]–[13].

With the trend of increasing computing power at the edge, federated learning finds applications in IoT. Mills et al. [4] addressed problems of federated learning like high communication costs and a large number of rounds for convergence. A new communication-efficient federated learning protocol

Workshop on Decentralized IoT Systems and Security (DISS) 2020
23 February 2020, San Diego, CA, USA
ISBN 1-891562-64-9
https://dx.doi.org/10.14722/diss.2020.23004
www.ndss-symposium.org

for mobile edge computing, which selects optimal clients for aggregating the updates in the scenario with limited resources, was presented in [5]. Wang et al. [6] proposed a control algorithms for edge computing architectures that adapts global aggregation frequency to minimize the learning loss. In [14], an architecture for smart home, focused on security and privacy and based of federated learning approach, was proposed.

Several approaches for federated learning can be distinguished. The aggregation can in principle happen in two manners. In *sequential* learning, sometimes referred to as (cyclic) incremental learning, computing nodes alternate in training the model, often in several cycles or rounds. Thus, each node receives the model at various stages of the training. In this case, each node trains the model for a certain subset of their data or a small number of training steps, before the model is passed on to others. In *parallel* aggregation, each node trains the model from all locally available data. The final model is obtained e.g. by averaging the model parameters.

Several types of attacks that exploit ML models to infer information about the training data have been discussed. *Model inversion* tries to recreate data samples that represent the underlying original objects. It has been shown to work in very specific settings, such as recreating pictures of persons to be identified by a face recognition system [15]. Ateniese et al. [9] developed an approach to extract statistical information about the training data from ML classifiers. They also showed that differential privacy is not preventing this leakage.

In this work, we focus on the *membership inference* attack [8], put in a sequential federated learning setting. This attack tries to predict if an instance was in the training set of a machine learning model, or not. It assumes that similar ML models trained on similar data must behave in an alike manner. Salem et al. [7] showed that membership inference is possible at a lower cost than was considered in the original paper [8]. They omit the assumptions about the adversary's knowledge of model architecture, data distribution and still preserve a high level of membership inference attack performance.

In [16], the authors differentiate between insider and outsider membership inference attack in *parallel* federated learning settings. In the former, an intruder has access only to the final model, while the *insider* is one of the federated learning parties, or the central coordination node, if existing. They show that federated learning is more vulnerable to insider attacks. The insider may be interested in not disturbing the outcome of the learning (as they would profit themselves from a high-quality model learned if the model is utilized to improve their services and experience), but are still interested in learning properties about other participants in the federation (e.g. because they are competing, or simply just curious).

In our setting, we further show that an insider can profit from the dynamics of models changing in the sequential (incremental) setting, where the insider can exploit the fact that the model is the most adapted to the data of the node it has just been trained on.

III. Experimental Setup

We briefly describe the setup for our evaluation.

Dataset: We use the *Purchases* dataset, which is a smaller version of the "Acquire Valued Shoppers" dataset from Kaggle[1] used in similar studies ([8], [16]). 600 binary attributes represent different products and show if an individual purchased the product. To obtain a classification task from this dataset, we follow the same procedure as [16]: we use a subset of $10,000$ records from the dataset, and use k-mean clustering to cluster the data with different numbers of target clusters, to obtain 10, 20, 50 and 100 classes. The resulting classes are generally rather similar in size, but a number of very small and large classes appear as well. Each class represents the purchase behavior of a group of customers, therefore the classification task is to predict the behavioral group for a specific customer.

Federated Learning and Model Architecture: We consider sequential federated learning (cyclic incremental learning) with 3, 10 and 15 processing nodes. For the learning step, we distribute the data equally among the different number of processing nodes, i.e. with three nodes, each party has $3,333$ instances, with ten nodes $1,000$, and with 15 nodes 666. We perform random sampling for this distribution. As a model for predicting purchase behavior, we use a neural network, as this model type is generally learned via an incremental optimization, e.g. gradient descend, and is thus a natural fit for sequential federated learning. Specifically, our network has one hidden layer consisting of 128 neurons, $tanh$ activation functions, a dropout layer, and a *softmax* layer. The learning rate is 0.001. We initialize the target model with random weights, and sequentially train it at each party with several iterations on the whole federated setting, a so-called cycle *(C)* (or round), following the same order (sequence) at each round.

Membership Inference Attack: The goal of the membership inference attack is to learn if a specific data instance was in the training set of the target model. The following description is based on [8]. Let D^{train}_{target} be the data on which a *target model* f_{target} was trained. $(x^i, y^i)_{target} \in D^{train}_{target}$ is an instance of the f_{target} training set, where x^i is an input vector and y^i is the output label. c_{target} is a probability vector and output of the target model f_{target}.

Essentially, membership inference is a binary classification task: for an instance (x, y), predict if it was "in" or "out" of D^{train}_{target}. To solve this task we can utilize a classifier f_{attack} called the *attack model*. The input of the attack model consists of a probability vector of size c_{target}, plus the correct label. The output of the attack model is whether the record was in the training set (labeled 1), or not (0).

The assumption underlying the membership inference attack is that similar models should behave similarly on similar types of data. We thus assume that we can emulate the behavior of the target model by training several *shadow models*. From these, we obtain the training set for the attack model.

In the insider threat model, the adversary knows the architecture of the target model, and thus can train the shadow

[1] https://www.kaggle.com/c/acquire-valued-shoppers-challenge/data

TABLE I: Membership Inference attack on centralized data

Dataset	Training Accuracy	Test Accuracy	Attack Accuracy	Attack Precision	Attack Recall	Attack F1
Purchase-10	0.958	0.781	0.614	0.571	0.913	0.703
Purchase-20	0.974	0.726	0.653	0.607	0.871	0.715
Purchase-50	0.992	0.645	0.728	0.659	0.943	0.776
Purchase-100	0.990	0.567	0.771	0.705	0.933	0.803

models with the same (or very similar) settings. We train a shadow model f^j_{shadow} on a dataset $D^{train}_{shadow_j}$, which is similar to D^{train}_{target}, but disjoint to it. For each $(x^j, y^j)_{shadow}$, a record in the dataset $D^{train}_{shadow_j}$, we get the probability vector $c^j_{shadow} = f^j_{shadow}((x^j, y^j)_{shadow})$. The tuple (c^j_{shadow}, y^j) is the input for the attack model with label 1. In the same manner, we obtain probability vectors for instances which are not in the training set of the shadow model and label them 0.

Membership Inference in Federated Learning: Considering sequential, incremental federated learning, an interesting attack model is having malicious users among the participants of the federation. If the adversary is an outsider, she has access to the final model. Even if she can infer that some records were in the training set, she is not able to distinguish in which particular party's set. But assuming that the adversary is part of the federated learning process, she can observe the dynamics of the learning, which might help in the inference task.

In our experiments, we train 10 shadow models of the same architecture as the target model, on the part of the Purchases dataset that was not used in the federated training by any of the participating parties. The attack model is a neural network with one hidden layer of 64 neurons and a *Sigmoid* activation function. The membership attack is simulated after each party trained the target model. We measured the accuracy of the target model's predictions for the training set and test set. As training set, we considered the training set of each node, but also evaluated the results for the union of all training sets from all federated learning parties. We also performed the membership inference attack on the training data, hence attacked each node. For the attack evaluation, we measured the accuracy, precision and recall of the attack model on the attack test set. The latter is a combination of target model predictions (probability vectors) for the data which was in the training dataset of a node, and the data which was not in the training set. The ratio of "in" and "out" data is 50%.

IV. Evaluation

For validating the effectiveness of federated learning itself, we compare the accuracy of the target model on the test set in federated learning with learning a model on centralized data. We maintain, and in some cases even reach higher accuracy score with all considered combinations of node numbers in federated settings. Therefore we conclude that federated learning does not deteriorate the quality of the target model.

As a baseline for the attack evaluation, we measure the performance of the membership inference attack on the model trained on *centralized data*. Table I shows the results of the membership inference attack for tasks with different numbers of classes in the training data. With increasing numbers of classes, the classification task becomes more difficult – therefore the gap between training and test set accuracy increases. With a higher number of classes, the accuracy and precision of the attacks increase as well. Our baseline attack results are comparable to those in literature ([8], [16]).

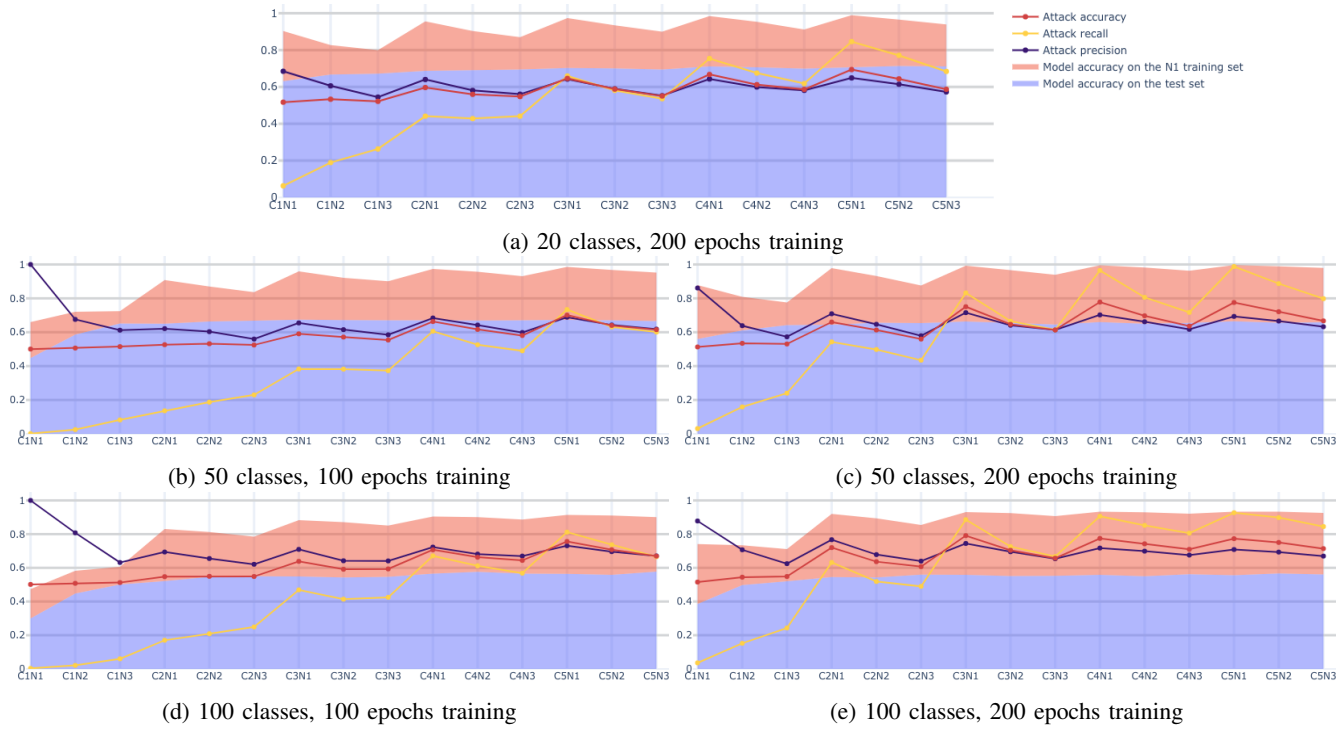
The attack accuracy depends in particular on the number of classes in the classification task, as well as the shadow training data, and the model overfitting to the training set, i.e. if there is a particularly high accuracy of the model on the training data and low accuracy on the test set [8]. The higher the number of classes, the more a model is extracting particular features from the training data to be able to distinguish between classes.

For evaluating membership inference in the sequential federated learning setting, we attack the target model after training at each node, i.e. in each cycle, we perform several, distinguished attacks. Figures 1 to 3 show the membership inference attack on the training data of the *first* node, $N1$. After the target model is trained at each of the following nodes, we measure the attack performance on the attack test set. Half of the attack test set consists of training data from the node to be attacked, i.e. $N1$ in this setting, the other half is test data that was not used for training in any of the nodes.

The vertical axis of the plots shows the score of the target model (Accuracy on node $N1$ training set, accuracy on the full test set) and the attack model scores on the attack test set (attack accuracy, precision, recall). The horizontal axis shows the target model state, where C stands for a federated cycle, and N stands for the node number. Thus $CiNj$ is the target model after training for the i^{th} cycle at the j^{th} node.

Figure 1 shows results for a membership inference attack in a sequential federated learning setting with three nodes. For the classification task with 20 classes (Figure 1a), attacking one node, we reach a higher attack accuracy than with training on centralized data. Moreover, starting from the second cycle of federated learning, the results show that right after training the model at the first node $N1$, the attack on $N1$ training data performs better at accuracy and precision scores. Figures 1b to 1e show how more training, which leads to an over adaptation to the specific training instances, influences the membership inference attack performance. With 200 epochs, for both the 50 classes and 100 classes tasks, the attack accuracy increases strongly right after training at node $N1$. This trend is way less pronounced, or not presented at all, in the case of 100 epochs training at each node, as the model does not have enough training time to adapt that strongly to the particular characteristics of the data available at each node. In the classification task with 50 classes, we reach membership inference accuracy around 0.8 (see Figure 1c), which is 7% higher than the attack on the corresponding model trained with the centralized data.

For federated learning on ten nodes, depicted in Figure 2, we can observe similar trends. Starting from the second federated cycle $C2$, right after training in the node $N1$, the membership inference attack on its data has better accuracy, precision and recall, which is subsequently declining after training the target model at each following node. Interestingly, the recall



(a) 20 classes, 200 epochs training

(b) 50 classes, 100 epochs training

(c) 50 classes, 200 epochs training

(d) 100 classes, 100 epochs training

(e) 100 classes, 200 epochs training

Fig. 1: Membership Inference in Federated Learning with three nodes. Attack performance during five federated learning cycles $C_1 - C_5$: precision, recall and accuracy $[0..1]$

TABLE II: Membership Inference attack on federated data on the last federated cycle

100 epochs training per node

Dataset	Test accuracy			Attack accuracy		
	3 nodes	10 nodes	15 nodes	3 nodes	10 nodes	15 nodes
Purchase-10	0.699	0.782	0.782	0.554	0.598	0.616
Purchase-20	0.700	0.720	0.702	0.577	0.636	0.654
Purchase-50	0.666	0.648	0.599	0.656	0.716	0.709
Purchase-100	0.578	0.527	0.471	0.713	0.724	0.700

200 epochs training per node

Dataset	Test accuracy			Attack accuracy		
	3 nodes	10 nodes	15 nodes	3 nodes	10 nodes	15 nodes
Purchase-10	0.726	0.768	0.771	0.594	0.604	0.615
Purchase-20	0.711	0.693	0.683	0.634	0.663	0.682
Purchase-50	0.659	0.599	0.589	0.713	0.708	0.714
Purchase-100	0.561	0.477	0.424	0.739	0.745	0.691

has a more drastic variation after training in the target node ($N1$). For all classification tasks, we reach higher attack accuracy attacking one specific node than in the centralized case. Similarly to the results with three nodes, 200 epochs on ten nodes causes better attack performance and higher variations in the attacked node. However, the accuracy of the target model is higher with 100 epochs per node than with 200, for all classification tasks. Nevertheless, even with 100 epochs for Purchase-100, we reach an attack accuracy of 88% when attacking node $N1$.

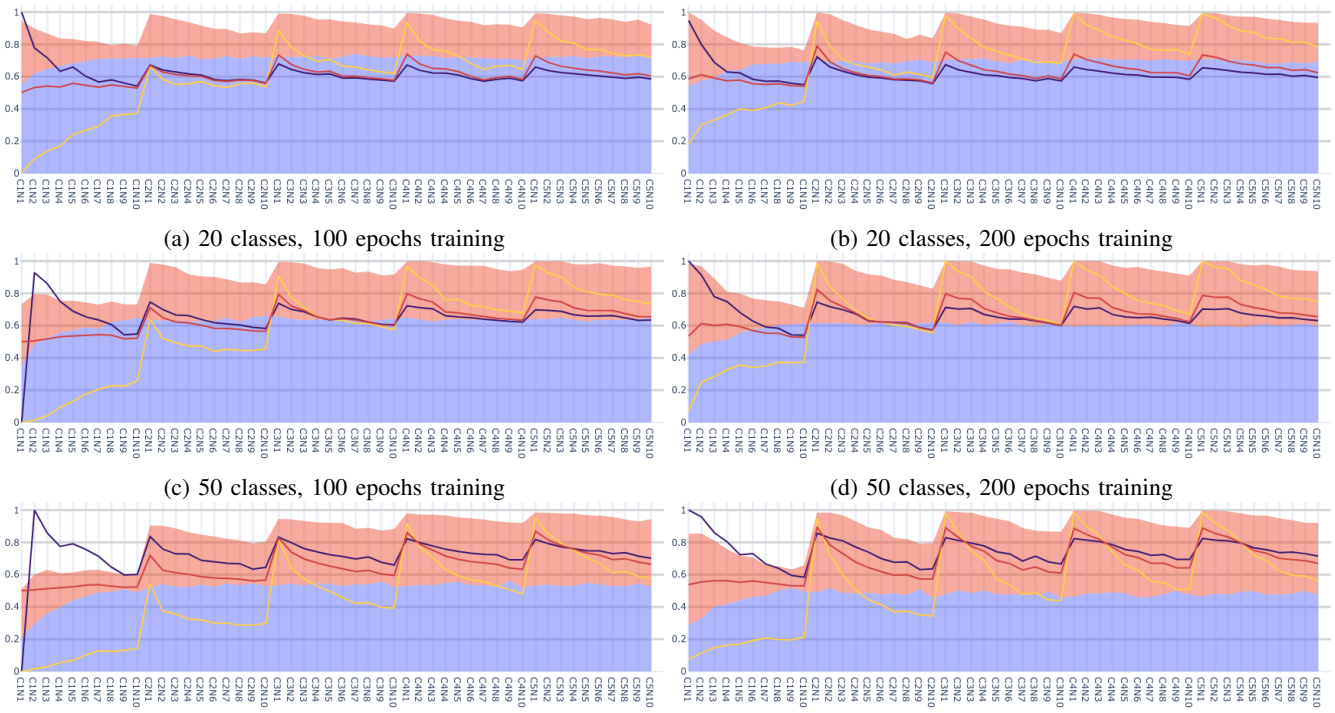
Federated learning with 15 nodes (see Figure 3) gives slightly different results on the target model trained with 200 epochs. Already in the first cycle, we can see the difference between membership attack performance at each federated learning node starting from around 60%. Figures 3b and 3f also show the trend that the attack accuracy is the highest right after training at the attacked node. Even on the Purchase-10 task, we reach an attack accuracy of 0.69. For Purchase-50 and 200 epochs, we reach 0.87 accuracy and 0.8 precision of membership inference from node $N1$. For Purchase-100, we reach an accuracy and precision of 0.92. However, the accuracy of the target model, in this case, is 10% lower comparing to centralized learning.

The membership inference attack works similarly when attacking the training data from other nodes: right after training in the target node the attack performance is better than at the neighboring ones in the sequence. Due to space limitations, these results are not depicted.[2]

Considering the scenario when an attacker has access only to the final model and wants to know if some instances were used by any of the nodes, we see in Table II that for Purchase-50 and Purchase-100, the attack accuracy is lower than in the centralized case. However, there are more risks if an intruder attacks some particular node and has access to the model before and after training in this node even at only one federated cycle. In this case, the attack on the model after training has significantly higher attack accuracy.

Summing up we find that attack performance is better on models which are trained more at each node, and the attack accuracy variation is increasing as well with a higher number of epochs. With a larger number of nodes, we generally also

[2]Data splits available at Zenodo, dx.doi.org/10.5281/zenodo.3667751



(a) 20 classes, 100 epochs training

(b) 20 classes, 200 epochs training

(c) 50 classes, 100 epochs training

(d) 50 classes, 200 epochs training

(e) 100 classes, 100 epochs training

(f) 100 classes, 200 epochs training

Fig. 2: Membership Inference in Federated Learning with ten nodes. Attack performance during 5 federated learning cycles $C1 - C5$: precision, recall and accuracy $[0..1]$

increase the attack accuracy, which can be explained by more overfitting, and a reduced number of instances per node.

V. Attack Mitigation Strategies

In this section, we briefly discuss mitigation strategies. The evaluation of their effectiveness for the setting of federated learning we discussed in this paper will be subject to future work. Some of the strategies apply to settings where the adversary is an honest-but-curious insider, and some of them can address also the setting where the coordinator, who receives potentially all messages, is the adversary. It is worth noting that, as with any defense mechanism in IT security, there is an associated cost, and in this case, the cost might mean computational overhead, and/or a decreased performance of the final model obtained (e.g. a lower prediction accuracy).

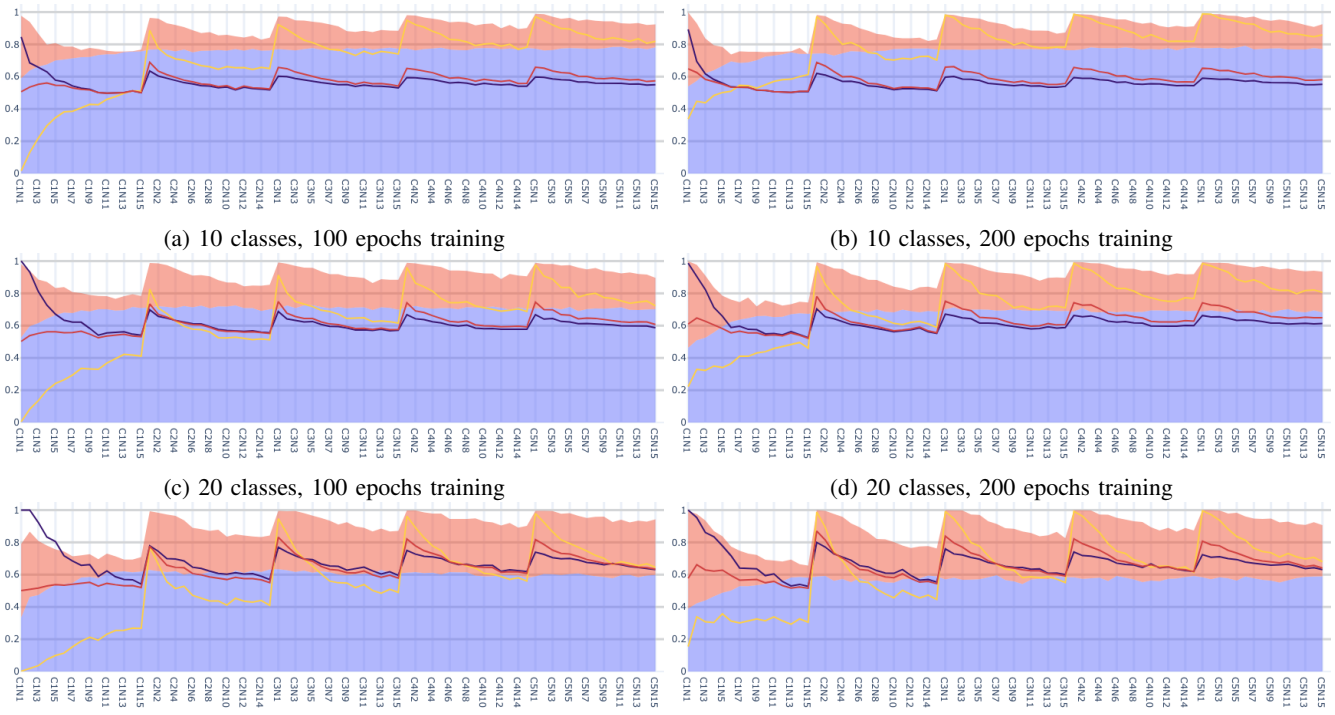
One approach for defense builds on the observation that an attack, right after the training a particular data set is used for, is beneficial for inferring the membership. We thus need to abstract the model from the specific training instances. This can be achieved by each node injecting a certain amount of noise to their training data to distort the resulting model. Another means to achieve a similar result is to apply differential privacy on the learning output, i.e. directly perturbate the learned model parameters.

To address insider attacks, randomizing the order of the nodes in each cycle can reduce the amount of information the insiders can obtain, as they do not necessarily know anymore when in the cycle a certain node was training the model, and they can thus not infer in detail of which participating node a data sample was a member. A distrusted coordinator could be limited in its actions by switching to a peer-to-peer model, where the nodes exchange the learned model directly with each other. Thus, the coordinator will only see the initial and final stage of the model in the sequential setting. In general, a lower number of training epochs at each node reduces the success rate of the attack. However, finding an optimal trade-off between the required amount of training to obtain an effective model, and at the same time adequately defend against membership attacks, is challenging. Instead, a higher number of cycles, each with a lower number of epochs per node, is a viable alternative to reduce the risk of exposure.

VI. Conclusion and Future Work

We investigated membership inference attack in sequential federated learning and showed that the attack in these settings can be more powerful. When there are multiple malicious participants of federated learning, they can observe the dynamics of the membership inference attack. As we showed, this is because right after training at the particular node, the membership inference attack on that node's training data has better accuracy. Therefore access to membership attack results from several nodes may bring additional information about the members' location at a specific node.

We also outlined possible mitigation strategies. Their evaluation for effectiveness, as well as their impact on the overall utility of the federated learning, will be subject to future work.



(a) 10 classes, 100 epochs training

(b) 10 classes, 200 epochs training

(c) 20 classes, 100 epochs training

(d) 20 classes, 200 epochs training

(e) 50 classes, 100 epochs training

(f) 50 classes, 200 epochs training

Fig. 3: Membership Inference in Federated Learning with 15 nodes. Attack performance during 5 federated learning cycles $C1 - C5$: precision, recall and accuracy $[0..1]$

ACKNOWLEDGMENT

This work was partially funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826078, project FeatureCloud, and the BRIDGE 1 programme (project KnoP-2D, No 871299) of the Austrian Research Promotion Agency (FFG),

REFERENCES

[1] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," Computer Science Laboratory, SRI International, Tech. Rep., 1998.

[2] L. Sweeney, "K-anonymity: A Model for Protecting Privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, Oct. 2002.

[3] C. Dwork, "Differential privacy," in *33rd International Colloquium on Automata, Languages and Programming (ICALP)*. Venice, Italy: Springer, July 2006.

[4] J. Mills, J. Hu, and G. Min, "Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT," *IEEE Internet of Things Journal*, 2019.

[5] T. Nishio and R. Yonetani, "Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge," in *IEEE International Conference on Communications (ICC)*. Shanghai, China: IEEE, May 2019, pp. 1–7.

[6] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive Federated Learning in Resource Constrained Edge Computing Systems," *IEEE Journal on Selected Areas in Communications*, Jun. 2019.

[7] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," in *26th Annual Network and Distributed System Security Symposium (NDSS)*. San Diego, CA: The Internet Society, 2019.

[8] R. Shokri, M. Stronati, and V. Shmatikov, "Membership inference attacks against machine learning models," *2017 IEEE Symposium on Security and Privacy (SP)*, 2016.

[9] G. Ateniese, L. V. Mancini, A. Spognardi, A. Villani, D. Vitali, and G. Felici, "Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers," *International Journal of Security and Networks*, 2015.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. FL, USA: PMLR, 2017.

[11] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, and S. Bakas, "Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation," in *International MICCAI Brainlesion Workshop BrainLes*. Granada, Spain: Springer International Publishing, 2018.

[12] A. Jochems, T. Deist, J. van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin, and A. Dekker, "Distributed learning: predictive models based on data from multiple hospitals without data leaving the hospital," *Radiotherapy and Oncology*, 2016.

[13] S. Silva, B. Gutman, E. Romero, P. M. Thompson, A. Altmann, M. Lorenzi, and U. K. Adni, "Federated learning in Distributed Medical Databases: Meta-Analysis of Large-Scale Subcortical Brain Data," Inria & Université Cote d'Azur, Sophia Antipolis, France, Tech. Rep., 2018.

[14] U. M. Aivodji, S. Gambs, and A. Martin, "IOTFLA : A Secured and Privacy-Preserving Smart Home Architecture Implementing Federated Learning," in *2019 IEEE Security and Privacy Workshops (SPW)*. San Francisco, CA, USA: IEEE, May 2019.

[15] 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 (CCS)*. Denver, Colorado, USA: ACM, 2015.

[16] S. Truex, L. Liu, M. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE Transactions on Services Computing*, 2019.