Data Poisoning in Sequential and Parallel Federated Learning*

Florian Nuding
florian.nuding@gmail.com
Vienna University of Technology
Vienna, Austria

Rudolf Mayer
rmayer@sba-research.org
SBA Research & Vienna University of Technology
Vienna, Austria

ABSTRACT

Federated Machine Learning has recently become a prominent approach to leverage data that is distributed across different clients, without the need to centralize data. Models are trained locally, and only model parameters are shared and aggregated into a global model. Federated learning can increase privacy of sensitive data, as the data itself is never shared, and benefit from the distributed setting by utilizing computational resources of the clients.

Adversarial Machine Learning attacks machine learning systems in respect to their confidentiality, integrity or availability. Recent research has shown that many forms of machine learning are susceptible to these types of attacks. Besides its advantages, federated learning opens new attack surfaces due to its distributed nature, which amplifies concerns of adversarial attacks.

In this paper, we evaluate data poisoning attacks in federated settings. By altering certain training inputs that are used in the training phase with a specific pattern, an adversary may later trigger malicious behavior in the prediction phase. We show on datasets for traffic sign and face recognition that federated learning is effective on a similar level as centralized learning, but is indeed vulnerable to data poisoning attacks. We test both a parallel as well as a sequential (incremental cyclic) federated learning, and perform an in-depth analysis on several hyper-parameters of the adversaries.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; • **Computing methodologies → Supervised learning**.

KEYWORDS

Federated Learning, Adversarial Machine Learning, Data Poisoning

ACM Reference Format:

Florian Nuding and Rudolf Mayer. 2022. Data Poisoning in Sequential and Parallel Federated Learning. In *Proceedings of the 2022 ACM International Workshop on Security and Privacy Analytics (IWSPA '22), April 24–27, 2022, Baltimore, MD, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3510548.3519372

*This work was partially funded from the European Union's Horizon 2020 research and innovation programmender grant agreement No 826078 (project 'FeatureCloud'). This publication reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.
SBA Research (SBA-K1) is a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the federal state of Vienna; COMET is managed by FFG

IWSPA '22, April 24–27, 2022, Baltimore, MD, USA

ACM ISBN 978-1-4503-9230-3/22/04…$15.00
https://doi.org/10.1145/3510548.3519372

1 INTRODUCTION

Machine Learning (ML) tries to extract value out of large volumes of data, enabling deeper insights by using e.g. classification models. Large amounts of training data are required especially in recent approaches, like (deep) neural networks. Often, data is collected at various different sites, independently of each other – frequently, however, it is is beneficial to analyze these scattered datasets together, to obtain a more effective model. An obvious approach is to centralize and jointly analyzing them in one device. But with growing security concerns, data privacy awareness and new data protection laws, e.g. the General Data Protection Regulation (GDPR), the demand for alternatives to centralized computation is growing.

A possible solution is Federated Learning (FL) [10]. Rather than centralizing data, it first creates models where the data resides. These intermediate models are then aggregated into a global model, e.g. through a central server or cryptographic protocols such as secure multi-party computation (SMPC). Thus, in FL, the data itself is always kept locally on the clients, and model generation is performed by the clients [9], thus also benefiting from the computing resources available there. Federated learning thus allows clients to benefit from each others' data, without explicit sharing the data.

Beside the many use cases for ML, several attacks on the ML process have been exposed. These can be categorized along the well-known CIA triangle (triad), which represents three dimensions: confidentiality, integrity and availability. Distributing the training process to potentially insecure or malicious clients creates additional attack surfaces that can be exploited, leading to attacks to each CIA dimension also in FL settings. Especially *adversarial examples* and *backdoor attacks* realized by data poisoning have a high potential in corrupting the intended use of a ML system [5].

Data poisoning modifies (parts of) the data used in the model *training* phase, and can affect both the integrity and availability of the model. By inserting manipulated data, an attacker tries to make the model learn the association of this pattern with a desired (wrong) label. At prediction time, adding this pattern to an input should trigger the model to predict the desired class. For example, in a backdoored face-recognition system, adding a certain pair of sunglasses to photos should always predict the person to be Michelle Obama. Ideally for the attacker, the model's behavior on benign data should not change, making the attack unnoticeable.

In this paper, we analyze the vulnerability of FL against data poisoning attacks. We consider a setting where data is gathered by multiple clients, and each client has a sizeable number of data records. This is referred to as cross-silo FL [6], as each client operates its own data silo: data is collected and remains decentralized. While cross-device learning can span up to millions of devices, e.g. mobile

phones, cross-silo learning generally deals with a lower-scale number of clients – [6] e.g. indicates 2–100 clients. For cross-silo FL, we analyze two paradigms: in *parallel* learning, each client simultaneously trains a local model, before sending updates to a central coordinator for aggregation. After this, a new round of learning may continue. In *sequential* learning, clients train in sequence, on the model that is the output of the previous client. This approach is suitable for a lower number of clients, e.g. a cross-silo setting with dozens of hospitals learning a common prediction model.

Our contributions are to study the effect of varying the (i) learning paradigm (sequential or parallel) (ii) number of clients (iii) backdoor pattern's prominence (iv) backdoor attack strategy (v) number of attackers, and (vi) timing of the attack. We perform an in-depth analysis of these effects on the success rate of poisoning attacks in FL on two different datasets, for traffic sign recognition and face recognition. We further provide a first in-depth analysis of poisoning attacks in sequential learning, and compare it to the better studied parallel learning, and publish two poisoned datasets.

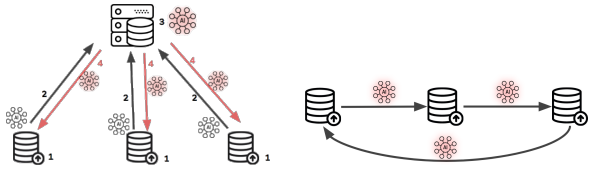
2 BACKGROUND AND RELATED WORK

We focus on the supervised machine learning task of classification, i.e. the process of determining a discrete label, and specifically on images, though the methodology is not limited to this type of data.

Convolutional Neural Networks (CNNs) are a Deep Learning method and the current state-of-the-art for analyzing images. They implicitly also learn representative features during model training. A neural network (NN) consists of neurons that are connected to each other by *weights* (the learnable parameters). A neuron (generally) receives numeric values as input, and forwards an activation by multiplying its own value by its outgoing weight to the next neuron. The architecture of a NNs consists of at three types of layers: an input layer, one or more hidden layers and an output layer. The input layer receives the data to be analyzed in numerical shape, while the hidden layer(s) consist of several neurons arranged in parallel and connected in series, by weights and biases. An activation function is applied to the output of a neuron, e.g. sigmoid functions or the Rectified Linear Unit (RELU).

When training a neural network, the current state of a model is used to obtain a prediction on the outputs of a *batch* of training samples. The predictions are compared to the correct labels (ground truth), and the resulting difference is a statistical estimator of the error gradient, which is used for updating the neurons' weights by a technique called *backpropagation*. In a backwards manner, the weights on each layer are adjusted by the gradient to minimize the prediction error. The amount of updating is also controlled by the *learning rate*. The number of samples used when calculating the error gradient is called *batch size*. While the maximum size of a batch is often limited by computational resource (such as memory), it is an important hyper-parameter of the model.

CNNs further include convolutional layers, which are useful especially in computer vision. Images are represented pixel-wise by a grid. In the convolutional layers, a relatively small $n \times n$ kernel matrix (e.g n=3, 5, 7) is "sliding" over the grid, filtering the image for noteworthy patterns (e.g. edges or shapes). It is usually followed by a pooling layer, which reduces the dimensionality of the input.



(a) Federated Averaging ("parallel") **(b) Sequential (incremental) federated learning**

Figure 1: Two approaches for Federated Learning

Federated Learning is a form of distributed model training. Rather than training on a central endpoint, it is performed on multiple clients, before being aggregated. Federated learning is used many different settings, from classification of medical images [19], to biomedical [21] and other structured data [17] to IoT settings In [12].

Horizontal learning [24] means that the clients' datasets has the same features, but differ in its observations (samples). In *Vertical* learning, the datasets share the same observations, but each client observes different features of them. E.g. in health care, the same patient takes different tests at multiple medical centers. We focus on horizontal learning.

We consider two approaches on how to combine local models. The likely most popular approach is a form of *parallel learning*, e.g. through *Federated Averaging*. All clients train in parallel, and the global model is obtained by averaging the local model parameters; training thus contains the following steps (cf. Figure 1a): (1) the coordinator distributes the current global model, (2) each client trains the model on their local data, (3) the clients send their models to the coordinator, which (4) combines the models to create a new, improved global model. These steps are often repeated in a number of cycles. From a security point of view, it is often assumed [15] that the coordinator is *honest-but-curious*, i.e. curious in extracting data of individuals, but honest in operations, and that the clients are honest in general – an assumption which can be exploited.

In *sequential learning*, also called *cyclic institutional incremental learning* [19], clients train locally, and then pass the current state of the model on to the next client in the sequence. Similar to the parallel case, learning is performed in a number of cycles, where each client trains once, as depicted in Figure 1b.

Adversarial Machine Learning subsumes several attacks, which can be categorized e.g. along the dimensions of the CIA triangle (triad). *Confidentiality* attacks in the domain of ML e.g. try to obtain the data used for training a model, and thus include Model Inversion [4] or Membership Inference [20]. These attacks typically happen after the model training. *Integrity* assures that a ML model is accurate, trustworthy and not altered. *Availability* guaranties a reliable and constant access to a model. Examples for attacks against these include *evasion attacks*, e.g. the well-known *adversarial examples*. With modifications to the samples to be predicted, they try to avoid correct outputs, but force a misclassification, potentially for malicious samples (e.g. trying to hide fraud, intrusions, or SPAM).

An attack during the training phase is *data poisoning*, shown already in [3]. In this attack, adversaries manipulate the training data, with the goal to alter the model's behavior in their own desire,

e.g. to force a specific misclassification. This behavior can be later exploited, and thus constitutes a backdoor. These attacks recently gained attraction for image classification, e.g. [5]. Here, the manipulation can e.g. be superimposing a specific "key" or pattern, e.g. a specific set of sunglasses in a face recognition system. While maintaining a high performance on the benign (original) test samples, all samples that contain the backdoor trigger shall be misclassified, e.g. to a predefined class – such as a user with high privileges in the setting of a facial authentication system.

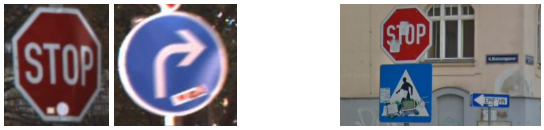
In [5], traffic signs are manipulated with e.g. a yellow square, flower or bomb, all resembling stickers glued to them. The authors concluded that more than 90-98% of the images with the backdoor trigger pattern were successfully predicted to the desired (wrong) target class, while the accuracy of non-backdoored images was still nearly as high as before. The authors of [23] show successful backdoor attacks in a variety of image classification benchmark datasets. They add white squared backdoors (which might be a very noticeable change) into the bottom right corners of the images. They make sure the patterns do not occur naturally in original images. On each tested dataset, they achieve a 97% success rate of the attack, while losing less than 3% accuracy on the benign data.

Besides adding a backdoor via data poisoning, another attack vector is to trick users to employ a previously learned and backdoored model via transfer learning. Transfer learning is a frequently used method for obtaining models for computer vision [5].

Adversarial Federated Machine Learning comprises e.g. inference attacks on the exchanged data [16]; our prime concern are data poisoning strategies in FL. The basic strategy follows a naive approach analogous to the centralized setting, and can be used for parallel as well as sequential learning. In parallel learning, benign and malicious clients train their models simultaneously, the malicious clients using also poisoned data. The learned parameters are sent as-is for aggregation to the coordinator. In sequential FL, when malicious clients receive the model, they train it with a fraction of poisoned data, and passe it on to the next client. In both settings, the success rate of the attack depends e.g. on the fraction of malicious clients; and the sequence of training in the sequential setting. Albeit using poisoned data, malicious client respect the aggregation protocol.

The *model replacement* strategy is a more advanced approach [1] for parallel learning. It increases the *magnitude* of the attacker's parameter updates, with the goal of the backdoor surviving the averaging process. Intuitively, this means that an attacker has to scale up the poisonous model's updates in relation to the number of clients that participate in the federated network, before it is sent to the aggregation process, ensuring that this update is more important than other, benign ones. The adversary may even try to fully replace the global model entirely by her malicious model.

The authors of [1] test backdoors in a federated averaging environment on two datasets: *CIFAR-10*[1] for an image classification task, and *Reddit* [11] for a word prediction task. They experiment with 10 clients in the federation, different model parameters, the basic and model replacement attack strategies, and three different shapes of backdoors. To achieve a backdoor success of 50% in the basic attack, 20% of all clients need to be malicious. To increase success over 90%, the attackers even have to make up more than 50% of the clients. For the model replacement attack, only around 1% malicious clients are needed for an accuracy of around 50%, and around 5% to reach the 90% accuracy threshold.



(a) Occurring within GTSRB **(b) Multiple stickers at a crossing**

Figure 2: Real-world traffic signs with "natural" backdoors

In [22], the authors also consider that benign clients contain data with the backdoor pattern, however correctly labeled. For IoT data, [13] evaluate backdoor attacks for federated averaging. For traffic signs, [14] evaluate both sequential and parallel learning.

3 EXPERIMENT SETUP

For our evaluation, we use the *German Traffic Sign Recognition Benchmark* (GTSRB) and the Yale Faces dataset. Both represent domains with practical ML applications, and have already been used for classification tasks, thus providing benchmark results to compare to. Moreover, GTSRB has already been used in literature addressing backdoor attacks (e.g. [14, 18, 23]) that we can compare to.

GTSRB consists of 50,000 photos of traffic signs in 43 different classes (types). The signs are not necessarily squared, and contain a border of around 10%. State of the art CNNs built by [2] are able to classify around 99% of the images correctly. The images are distributed very unevenly among the classes, from 200 to more than 2,000 samples per class. Image sizes vary from 15×15 to 250×250 pixels. For training our CNN, all images need to be of equal size. To be comparable with [2], we also resize all images to 32×32.

As [23] test backdoors patterns in the form of squares at a size of 1% of the image, located on the bottom right corner, we use squared size patterns as well. We also vary sizes (1% and 0.5% of the signs area) and colors of these backdoors. Via a script, we place the backdoors in an area of ±20% of the image width around the center of the images, to ensure that the patterns are on top of the actual traffic sign – patterns placed *outside* the sign as in [23] would have limited practical relevance, especially for physical world attacks. The dataset with backdoor patterns is available at Zenodo[2].

Note that the inconspicuousness of the pattern is important for the attack. In a real world setting, a highly suspicious backdoor will more likely be identified. While e.g. the superimposed squares are noticeable, they are chosen to be not suspicious, as they naturally occur in the selected data: a square "sticker" on a sign does not raise great attention – as many signs are already covered with stickers anyway. Examples of this are shown in Figure 2. We found many stickers actually exceeding our maximum size of an area of 1% by far. Also, they occur in a large variety of colors motives.

For the model architecture, different CNNs based on literature were tried. Research on several CNNs for classification of traffic sign datasets was performed in [2], and we finally implemented their best

[1] https://www.cs.toronto.edu/~kriz/cifar.html

[2] Dataset: DOI 10.5281/zenodo.3716766, resulting models: DOI 10.5281/zenodo.3723574

Table 1: Targets of different actors in the learning process

	normal client	**attacker**
benign input data	high accuracy	high accuracy
poisoned input data	low accuracy	high accuracy

performing CNN. It consists of two convolutional layers, followed by a max-pooling layer, and again two convolutional layers followed by a max-pooling layer. Then, three times, dropout regularization is performed, each followed by a fully connected layer. We employ a relatively simple architecture. However, backdoor attacks have been shown to be successful in more complex architectures as well [1]. Thus, our selection is not biased to a network more susceptible to attacks, and our conclusions are applicable to other architectures.

The **Yale Face** dataset [3] contains greyscale images of the faces of 15 individuals; all classes are evenly distributed. Each image of a person shows the person's face at a different facial expression, such as "happy", "normal", "sad", "sleepy", "surprised", "wink", different lighting, angle (from the center, left, or right), and with or without glasses. Each image is 285x224 pixels large; to be comparable to [7] we resize them to 64x64. For our experiments, we rebuild a state of the art model first used by [7], which classifies 80% of the images correctly. For data augmentation, we apply a horizontal flip and a change of brightness by ±60% and ±30% on the training set.

We generate different types of backdoors with the application "FaceApp"[4]. Due to the nature of this app (faces are analyzed automatically and patterns are fitted individually), the backdoor patterns are not entirely identical. These "non-identical but similar" patterns are useful to illustrate potential robustness of the resulting backdoors. The dataset and example patterns are available at Zenodo [5].

Our results are measured using effectiveness metrics on the test set, namely accuracy (overall and per class). Each measurement is done after a federated training cycle finishes. The goals of different actors are summarized in Table 1. A "normal" client aims for high accuracy on benign data, and poisoned samples should be classified as if they were not modified, thus result in a low backdoor accuracy. The attacker's goal is to increase the accuracy on backdoored (poisoned) samples, while keeping a high accuracy on benign input data to not raise suspicion. We report results from the viewpoint of the attacker – in other words, we consider a high accuracy as "good" on benign as well as backdoored data.

For the federated setup, we consider different numbers of clients; in most settings, we experiment with five, ten and 20 clients, which is comparable to literature, e.g. [1] uses ten clients.

4 RESULTS

Here, we present our results and an analysis therefore; we first discuss the GTSRB dataset, followed by the Yale Faces.

4.1 Traffic Sign Recognition

We first compare the behavior of federated to centralized learning, to confirm that we analyze a model that achieves results comparable to state-of-the-art. As mentioned, [2] analyze GTSRB in depth. Their best architecture, which we reuse (cf. Section 3), is trained for 75 epochs. At batch sizes of 5, 10 and 20, they achieve a test set accuracy of 99.35%, 99.01%, and 98.67%, respectively. We reuse this architecture, and to the best of our knowledge use the same parameters, and are able to reproduce their results. Due to the large number of experiments we perform, we subsequently however need to use a larger batch size. By increasing it to 512, we drop below 3% in accuracy, but gain a 72 times speedup (Figure 3a).

Regarding the *number of clients*, for two clients, sequential and parallel learning learning perform nearly identically over the 100 training cycles. For five clients, both methods again converge to the same level of test set accuracy; however, the sequential approach converges faster. With ten clients, sequential performs almost identical as with five. However, parallel converges much slower, and does not achieve the same effectiveness as with two or ten clients at 100 cycles, but needs significantly longer to reach it (Figure 3b).

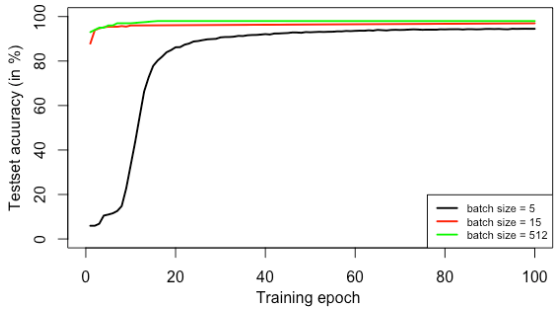
A *skewed, not independent and identically distributed* (non-iid) setting, where some classes are distributed only to some clients, has a large effect. We randomly selected 34 classes, representing roughly 80% of all classes, making up also around 80% of the total number of observation, and distribute them across all participating nodes. The remaining nine classes are present only at one client.

For sequential FL, if these non-iid classes are trained before the other classes in each cycle, they are learned slower, and partially stagnate at lower effectiveness. While classes known to all clients in average perform comparable to the iid case, there is a high variance in their performance, and some classes are taking longer to converge or do not reach the accuracy after 100 epochs at all. If the non-iid classes are trained at the end of the cycle, they reach a higher accuracy already in the early stages of training (Figure 4); the overall effectiveness at the end of the training is similar, though (Figure 5a). For parallel FL, non-iid classes take much longer to be learned, and do not reach the same accuracy (Figure 5b).

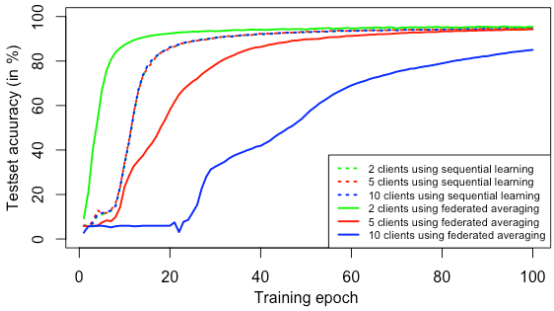
In conclusion, class distribution does have a significant influence on federated learning, which is relevant, as this is similar to a data poisoning attack, where poisoned data can be seen as a non-iid class. In sequential FL, the order of the classes is crucial. Training the non-iid classes in the end of a learning cycle indirectly acts as boosting the learning process on these samples, as they are represented to a higher degree in the final model. In parallel learning, data that is non-iid drastically reduces the overall effectiveness, and the performance of the "exclusive" classes increases later and slower; this effect increases with the number of exclusively known classes.

4.1.1 Timing of the attack (sequential learning). In sequential learning, attacks can happen at different points of time; we evaluate the attacker being the first or the last in sequence – which is similar to the non-iid classes being first or last. As shown in literature, an intrinsic property of sequential learning is *catastrophic forgetting* [8]: data that is learned in early stages of training is likely to have less influence on the result – it is "forgotten". In Figure 6, we depict the results on benign and poisoned test sets. There, the x-axes represent the "tested point of time", which corresponds to one evaluation on the test set. Thus, testing two times per cycle results in 200 test points. Red dots represent testing after the first set of clients (e.g.

[3] http://vision.ucsd.edu/content/yale-face-database
[4] https://www.faceapp.com/
[5] Dataset: DOI 10.5281/zenodo.3774167, resulting models: DOI 10.5281/zenodo.3774170

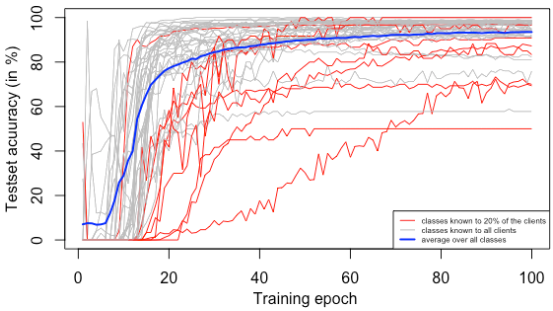


(a) Varying batch sizes (centralized learning)

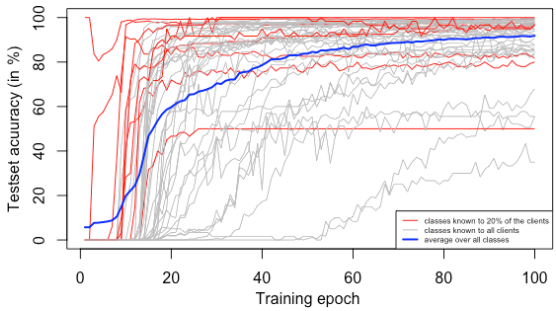


(b) Varying number of nodes (sequential vs. parallel)

Figure 3: Baseline evaluation of federated learning without attacks

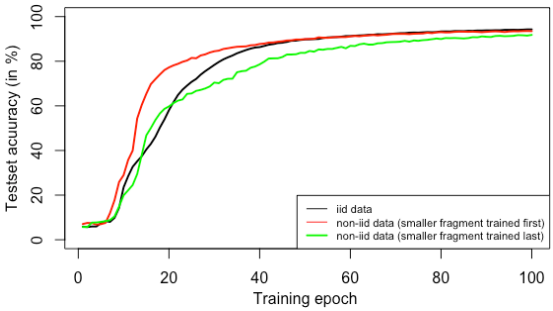


(a) Exclusively known classes first

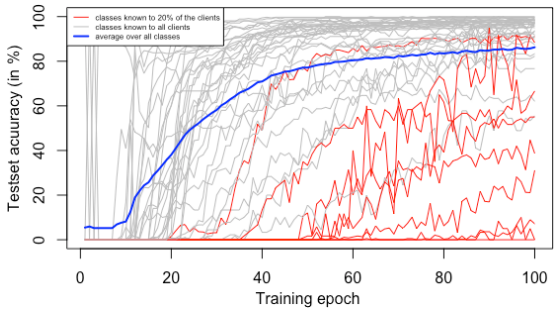


(b) Exclusively known classes last

Figure 4: Baseline evaluation of federated learning: timing of training on non-iid data, sequential



(a) Sequential learning



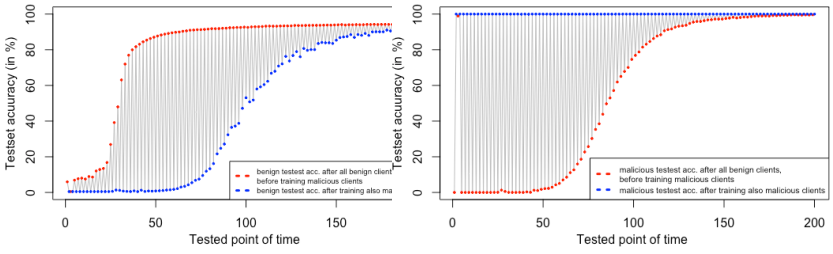
(b) Parallel learning

Figure 5: Baseline evaluation of federated learning: training on iid and non-iid data

all benign) have trained, and blue dots testing after the other set of clients (e.g. all malicious) have trained as well.

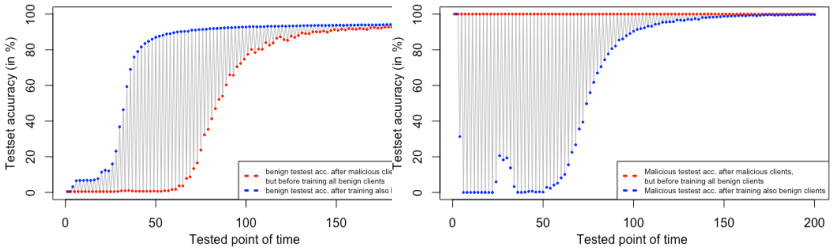
Figures 6a and 6b shows results when the attacker trains *last*. The performances on the benign test set (Figure 6a) shows the effect of the malicious client overriding the benign ones: while tested after the benign clients trained, the accuracy reaches more than 80% already after around 20 cycles (40 ticks). However, it takes around 75 cycles (150 ticks) to reach this level when a malicious client is trained afterwards. On the poisoned test set (Figure 6b) we can see that the backdoor is introduced immediately after the malicious client is trained, but it takes more than 50 epochs (100 ticks) to stay at a level of 80% accuracy after the model is again trained by the benign clients. Thus, data learned from earlier cycles seems indeed to be forgotten, especially in the first half of the training.

In Figures 6c and 6d we test the opposite case – the malicious client is trained first. We observe that the benign test set takes around 20 epochs (40 ticks) to reach an accuracy of 80%, while the backdoor takes around 45 epochs (90 ticks) to be successfully introduced. When tested right after the malicious client trained, the backdoor is introduced almost immediately, while the benign data takes more than 50 epochs to reach the 80% accuracy level. This again confirms the catastrophic forgetting. Comparing Figures 6a



(a) Malicious last, benign test set

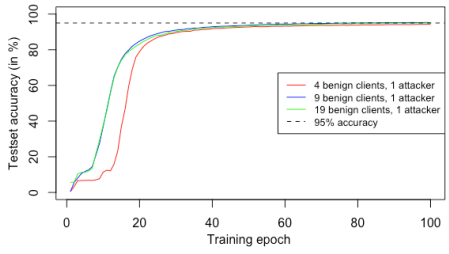
(b) Malicious last, poisoned test set



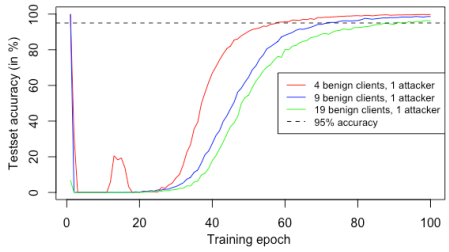
(c) Malicious first, benign test set

(d) Malicious first, poisoned test set

Figure 6: Traffic sign dataset, sequential learning, one malicious client and four benign clients



(a) Benign test set



(b) Poisoned test set

Figure 7: Varying numbers of clients (malicious first)

and 6b with Figures 6c and 6d, we can see that they are almost inverse of each other. When the attacker trains first, the model first learns to predict the benign data correctly; when the attacker trains last, this behavior is opposite.

We further tested with a total of ten and 20 clients, with always one adversary included, and obtain very similar results. Figure 7 depicts this for the attacker training first. For ten and 20 clients, the outcomes are nearly identically, while for five clients, it takes ten epochs longer to reach a converging state on benign data. On the malicious dataset, having only four benign clients trained after the malicious clients leads to the fastest introduction of the backdoor pattern. But also in federations with nine and 19 benign clients, the adversary is able to introduce the backdoor with 95% accuracy, having to train two respectively ten epochs more. For the adversary training last, a different number of clients results in an almost identical accuracy on both test sets (not depicted).

These observations lead to the following conclusions. (i) If the training phase is long enough, the sequence of the participants does not matter. (ii) If training is short, it is important for an attacker to train last. This observation can be exploited for a defense against backdoors in sequential learning. By placing some trusted clients at the end of the training cycle and stopping the training at a point when the overall training accuracy converges, one might prevent adversaries from entering the backdoor.

4.1.2 Attack Strategies (federated averaging). We attack parallel learning by the "basic" and the model replacement strategy. From Figure 8, we observe that the backdoor insertion into the global model using the *basic* strategy performs best when all available data in the malicious clients is poisoned. In this case, the accuracy reaches 80% after 100 cycles, after staying close to 0% for the first 60 cycles. When training longer, the accuracy on the poisoned test set steadily increases. It eventually exceeds 95% test set accuracy after 145 cycles for attackers using 100% poisonous data.

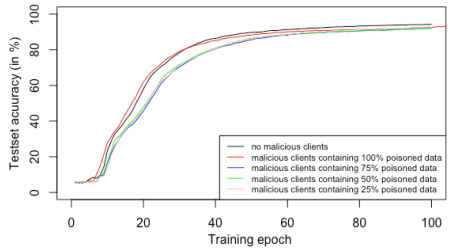
The success rate of the attack drastically reduces if the number of clients in a federated network increases, starting already with ten clients. With 20 or more clients, the backdoor accuracy drops to under 1%, making this strategy unusable (not depicted).

For *model replacement*, a very important parameter is the fraction p of poisoned samples in malicious clients. As this strategy to some extent replaces the global model with the attacker's local model, this ratio acts a a trade-off between accuracy on the benign vs. accuracy malicious test data – to be successful also on benign data, malicious clients need to train with benign observations as well. In Figure 9b we report results for p between 100% (only poisoned samples) and 0% (no poisoned samples). For this setting, the optimal value for p appears to lie between 25 and 50%. This is in agreement with literature, as e.g. [1] used a percentage of 31% at 5% poisoned clients to reach a backdoor accuracy of over 90%.

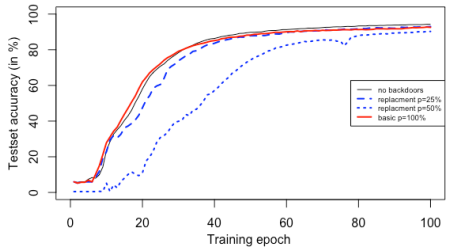
In conclusion, *model replacement* clearly outperforms the basic attack strategy. Our best performing value in the basic attack strategy (p=100%) reaches a lower level of backdoor accuracy, and also needs more cycles to become successful (see Figure 8d). Furthermore, the more clients are participating in the learning, the larger the performance difference between the two methods becomes.

4.1.3 Appearance of the backdoor. Following prior work [5, 23], we use a squared pattern of 1% size of the original image in green, as green is rarely present on traffic signs. In contrast, black frequently appears on traffic signs, and we thus include it with a size of 1% as well. The third backdoor pattern, a green square of size 0.5%, is tested to gain an insight on the importance of the backdoor's size.

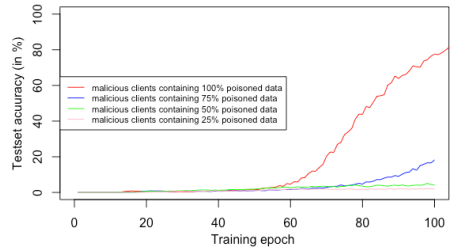
We discuss the influence of different patterns in federated averaging, with the model replacement strategy, for four benign and one malicious client, to achieve a targeted 95% backdoor and benign



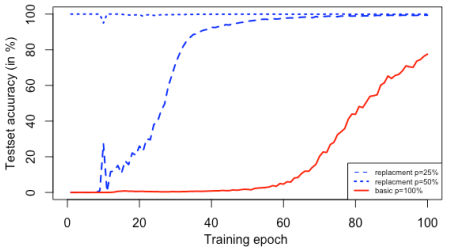
(a) Basic attack, different % poisoned, benign test data



(b) Basic attack vs. model replacement, benign test data



(c) Basic attack, different % poisoned, malicious test data



(d) Basic attack vs. model replacement, malicious test data

Figure 8: Basic attack vs. model replacement (four benign, one malicious clients)

accuracy. For the smaller 0.5% green sized backdoor in Figure 9a, we can see that backdoor accuracy at 25% and 50% poisoned data oscillates around only 10% and 50% at the end of the training, and is thus less successful. If the attacker uses 75% poisoned data, the backdoor is successfully introduced – but at cost of performance on the benign test set, which drops to 43%.

Figure 9b shows results for the 1% sized green pattern. We can observe that when attackers use 25% and 50% poisoned data, the accuracy on the benign as well as the poisoned test set exceed the 95% threshold after 100 cycles – the attacks are successful. We further see that black color (cf. Figure 9c) is less effective: the only attack considered successful is with 50% poisoned data. If the attacker uses less, the accuracy on the poisoned test set is too low.

To summarize, we can see an impact of the pattern's size as well as color on the success rate. Larger backdoors work better, and the (in this dataset) rarely occurring light green color works better than the very common black. Based on these observations, recommendations for an attacker are thus to select a larger and unusually colored backdoor pattern to increase the chance of the attack to be successful, avoiding too suspicious patterns (cf. Section 3)).

4.1.4 Ratio of malicious to benign clients. Figure 10 shows the effect of the ratio of malicious to benign clients on the global model's accuracy for sequential learning when attackers train last. We can observe that all tested values lead to an accuracy of over 95% both on the poisoned and benign test data. While the accuracy on the benign test set increases notably slower than if there were no attackers, the backdoor is effective from the first cycles of training in all tested cases. A similar observation can be made for training the malicious clients first (not depicted) – also here, all tested values reach 95% accuracy on both test sets at similar times.

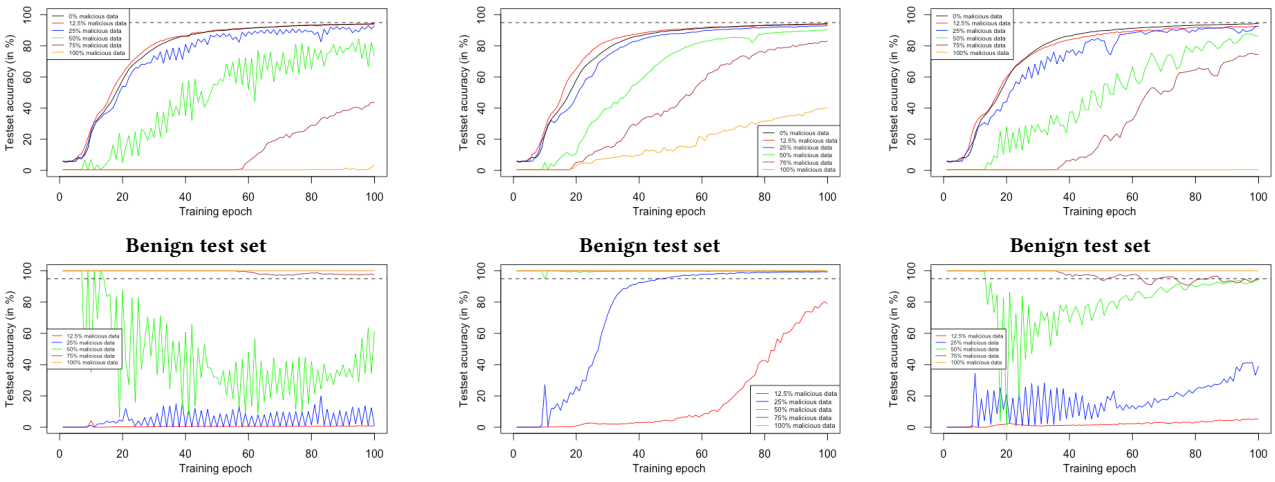
In parallel learning with the basic attack strategy, the best results were achieved with clients containing 100% malicious data (cf. Section 4.1.2), which we thus use. In Figure 11 we compare having one (red), two (blue) or three (green line) attackers among five clients (the rest are benign). We conclude that a higher ratio of malicious clients leads to a better insertion of the backdoor into the global model. However, a more important factor is the *absolute* number of clients. Fixing a malicious client rate of 20%, we observe that in a federation consisting of five clients the backdoor is introduced more successfully than in a federation of ten clients (not depicted).

Regarding model replacement, we tested cases with four benign and one malicious (Figure 9b) and nine benign and one malicious client (not depicted). In both cases, the backdoor is successfully introduced at an accuracy of $> 95\%$ in the case of $p > 50\%$. On the benign test data, in general, learning with a higher number of participants reaches high accuracy slower (cf. Figure 12), which is in line with the observations made in general that federated learning with many clients takes longer to converge.

To summarize, in sequential learning, the effect of the fraction of malicious clients is relatively small: all tested ratios (5% to 20%) lead to a successful introduction of the backdoor. This fraction, however, has a significant impact on parallel learning: for both attack strategies, a larger ratio leads to a faster backdoor insertion. Notably, similar to observations by [13], the effectiveness of the attack is not simply connected to just the fraction of poisoned data among the clients. Instead, if the same amount of poisoned data is distributed among multiple attackers, the attack is more successful.

4.2 Face Recognition

We start with a detailed evaluation of the beard pattern's capacity to be used for attacks in sequential FL. In Figure 13a, per class accuracies are shown. Black bars represent the base line, a federation without any attacker, and gray represents a sequential federation of 4 benign clients and 1 attacker that trains at the end of the learning cycle. While the accuracy stayed the same and decreased for some classes, it is noteworthy that it *increased* for classes 1, 3, 4, 5 and 10.

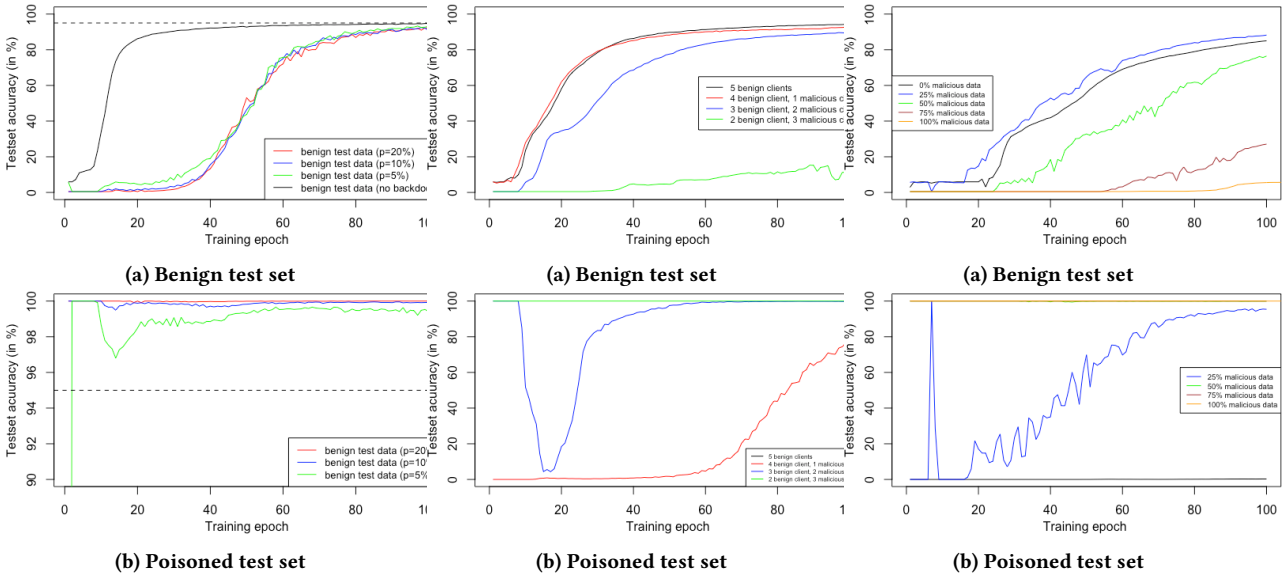


Benign test set **Benign test set** **Benign test set**

Poisoned test set **Poisoned test set** **Poisoned test set**

(a) 0.5% sized green pattern **(b) 1% sized green pattern** **(c) 1% sized black pattern**

Figure 9: Backdoor success, basic attack strategy in parallel federated learning



(a) Benign test set **(a) Benign test set** **(a) Benign test set**

(b) Poisoned test set **(b) Poisoned test set** **(b) Poisoned test set**

Figure 10: p% adversaries (sequential, adversaries last) **Figure 11: Number of malicious clients (basic attack)** **Figure 12: Scenario from Figure 9b at 8 benign clients and 2 malicious clients**

We believe that this unexpected behavior is a result of overfitting in the not-attacked network, which converges after only ten epochs, and for which the overall accuracy drops from 95% to 92% over the following 90 epochs. By adding malicious data this overfitting might be decreased for these classes, as the convergence is slower.

The red bars of Figure 13a represent the accuracy of the poisoned data, i.e. the amount of images that are misclassified as intended by the attacker (n.b. that classes 5, 7, 11 and 12 were not selected for adding a beard, as the individuals already wear a beard.). In average, the attack is successful by 87%, while it under-performs for classes 9 and 13. Figure 13b shows the confusion matrix of the poisoned samples (n.b. that if all poisoned samples were successful, each would be matched to target class 1 in the x-axis). Most poisoned samples are indeed classified as intended to the target class 1, or remain in their original class, i.e. there is only minor untargeted misclassification. Notably, we observe that four samples from class 9 are misclassified as class 7 (instead of the targeted class 1), a person naturally carrying a full-beard.

We are able to establish the beard backdoor to a high degree, while only dropping accuracy by a few percent on the benign test set.

As with the traffic sign dataset, the later an adversary trains in the cycle, the more successful the attack is – but then, more degradation happens on the benign test data. A larger ratio of benign clients, as well as larger federation in general, leads to reduced attack success.

For parallel learning, we can observe a similar behavior as with the traffic signs – the basic attack strategy results in at most 30% intended misclassification, and can thus not be considered successful.

In the model replacement strategy, we demonstrate the impact of the ratio of benign to poisoned data, which was very relevant to the success on GTSRB, for a federation with 4 benign and 1 malicious client (Figure 14) We can see that with 12.5% and 25% poisoned data, the accuracy on the benign test data is very close to the one without attacker, but the backdoor success is only around 35% and 44%, respectively. With 50% poisoned data, the accuracy on the benign test is 72%, a 20% drop compared to no attacker, while the backdoored success reaches 82%. Thus, there is a clear trade-off for the fraction of poisoned data of the attackers. The higher, the better the backdoor is implemented into the model, but the worse it performs on benign data. This trend continues when increasing the fraction to 75% and 100% poisoned data. With more clients, e.g. with 9 benign and 1 malicious client as in Figure 15, we observe the same trend – a higher fraction leads to a better backdoor, but costs prediction performance on benign data. In general, the performance on the benign test data is lower with more clients, also when no attacker is present. The model replacement attack, in contrast, performs at comparable levels in both settings.

In Figure 16a, we detail the per class accuracies for this case (parallel learning, 4 benign clients and 1 attacker, model replacement strategy) for 25% poisoned data. Overall, the backdoor is introduced at an accuracy of 45%, while benign test data reaches an accuracy of 88%. We clearly see that most of the data is still classified as if the backdoors was not present, observable by the non-negative values in the diagonal. Figure 16b shows per class accuracies at a 50% fraction of poisoned data – now, most (83%) of the poisoned data is correctly misinterpreted as class 1. The drop to 72% on the benign test set mainly results from benign samples also classified as the backdoor target, class 1.

For the glasses pattern, we observe very similar behavior of the backdoored classifiers as with the beard pattern, however, most of the times at a slightly lower level. This is likely due to the larger size and thus more prominence of the full beard. Due to space limitations, as a representative example, we illustrate sequential learning with 4 benign and 1 malicious client. The overall backdoor accuracy reaches 88%. As seen from the per class accuracies in Figure 17, the biggest outlier is class 7 – the attack does not work for this class, and the test data is still classified as if no attack was carried out. Also, even tough our glasses pattern is similar to the glasses that are already worn by person 8, there are only two poisoned samples that are wrongly predicted to this class.

For parallel FL and the model replacement attack with 25% poisoned data, the backdoor succeeds only at 25%, comparable to the beard pattern. Especially for an attack on a biometric authentication system, this would be considered too low. The backdoor success increases to 81% when using 50% of poisoned samples – however, the overall accuracy of the benign test set dropped to 52%. This is mainly due to the classes with a high success rate on poisoned test set resulting in a lowered accuracy on the benign test set.

Generally, we observe that we can introduce the glasses backdoor into the global model, although at a higher cost than with the beard pattern: the benign test set accuracy drops from 81% to 52%, instead of from 82% to 72% – at a comparable attack success on the poisoned test-sets. This might be due to the glasses pattern in general constitutes a smaller area compared to the beard backdoor.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we provided a detailed analysis of the impact of data poisoning attacks in federated learning. We considered two different approaches to the federation (parallel and sequential), and evaluated a number of poisoning variations on two different datasets, for traffic sign recognition and face recognition.

The appearance (color and shape) of the backdoor pattern influences the success rate of the adversary, and should therefore be carefully chosen. Larger patterns, and starker contrast increase the success; the adversary shall thus chose such patterns, and as real-world examples showed, these might still be chosen to appear unconsciously.

For sequential learning, the effect of catastrophic forgetting has also an impact on the adversary, depending on the order of when the malicious data is presented to the model. However, in general, if sequential learning is performed for enough cycles, it offers the best chance for the attacker to reach both a high accuracy on the backdoored samples, as well as maintaining good results on the benign data, i.e. being less noticeable. While sequential learning has advantages such as e.g. not needing a coordinator, this susceptibility to stealth attacks can be seen as a major drawback.

In some settings, the adversary needs to tailor the training process to be successful, e.g. using the model replacement strategy for (parallel) federated averaging, especially when there is a larger number of clients. While there were further differences in the effectiveness and speed of embedding a backdoor depending on the setup of the federation, like the number of overall participants, and the percentage of poisoned samples, also the parallel federated learning process has shown to be vulnerable to data poisoning attacks aiming.

Our observations and conclusions are very similar for both traffic sign and face recognition data, and different types of backdoors, and thus likely generalize well to other domains. Considering that the federated system is a distributed one, and the multitude of participants likely offers easier options for an adversary to manipulate one node, the power that this one node receives over the training process is a reason for concern. Therefore, future work needs to specifically address the issue of defending against such attacks in a federated learning setting.

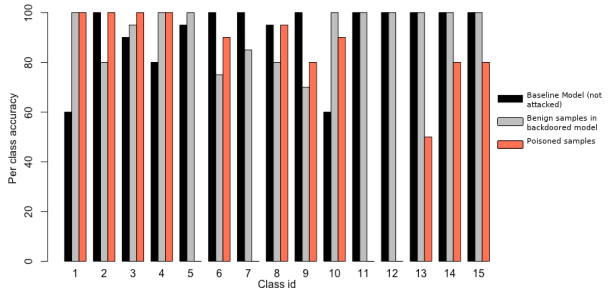
REFERENCES

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How To Backdoor Federated Learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, Palermo, Italy.

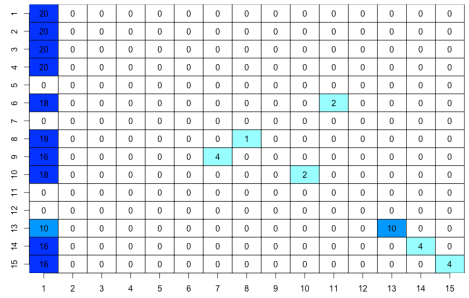
[2] Josef Bengtson, Filip Heikkilä, Per Nilsson, Lukas Nyström, Erik Persson, and Gustav Tellwe. 2018. Deep learning methods for recognizing signs/objects in road traffic.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning Attacks against Support Vector Machines. In *International Conference on Machine Learning (ICML)*, John Langford and Joelle Pineau (Eds.). Omnipress, New York, NY, USA.

[4] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures. In

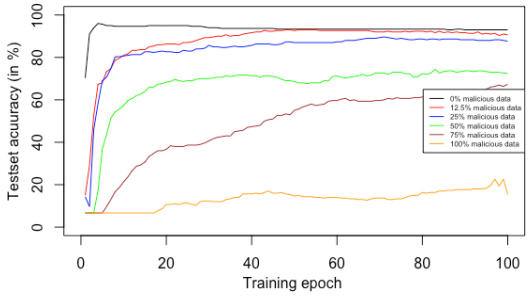


(a) Per class accuracy

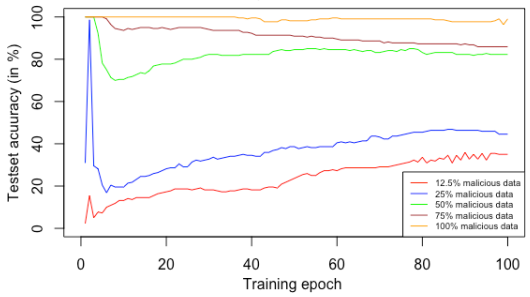


(b) Confusion matrix of the backdoored samples

Figure 13: Sequential learning, 4 benign and 1 malicious client, adversary last, *beard* pattern

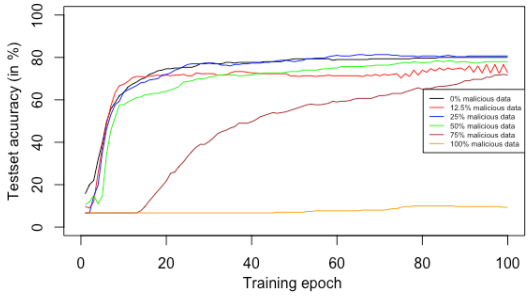


(a) Benign test set

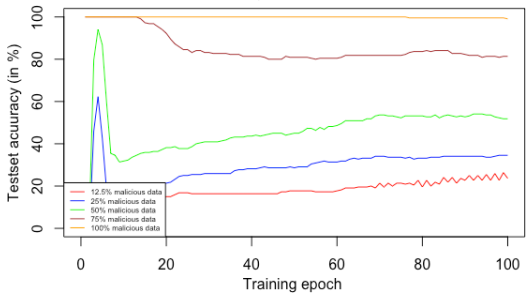


(b) Poisoned test set

Figure 14: *Model replacement*, 4 benign and 1 malicious clients, beard pattern



(a) Benign test set



(b) Poisoned test set

Figure 15: *Model replacement*, 9 benign and 1 malicious clients, beard pattern

ACM SIGSAC Conference on Computer and Communications Security (CCS). ACM, New York, NY, USA. https://doi.org/10.1145/2810103.2813677

[5] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. BadNets: Evaluating Backdooring Attacks on Deep Neural Networks. *IEEE Access* 7 (2019). https://doi.org/10.1109/ACCESS.2019.2909068

[6] Peter Kairouz, H. Brendan McMahan, et al. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning* 14, 1–2 (2021). https://doi.org/10.1561/2200000083

[7] Hurieh Khalajzadeh, Mohammad Mansouri, and Mohammad Teshnehlab. 2013. Hierarchical structure based convolutional neural network for face recognition. *International Journal of Computational Intelligence and Applications* 12, 03 (2013). https://doi.org/10.1142/S1469026813500181

[8] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America* 114, 13 (2017). https://doi.org/10.1073/pnas.1611835114

[9] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. 2016. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. arXiv:1610.02527

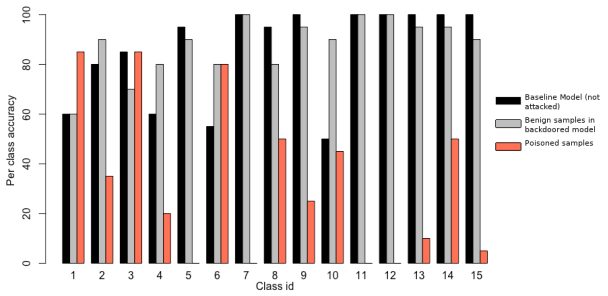
[10] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated Learning: Strategies for Improving Communication Efficiency. In *Workshop on Private Multi-Party Machine Learning, Conference on Neural Information Processing Systems (NIPS)*.

[11] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *International Conference on Artificial Intelligence and Statistics*. PMLR, Fort Lauderdale, FL, USA. http://proceedings.mlr.press/v54/mcmahan17a.html

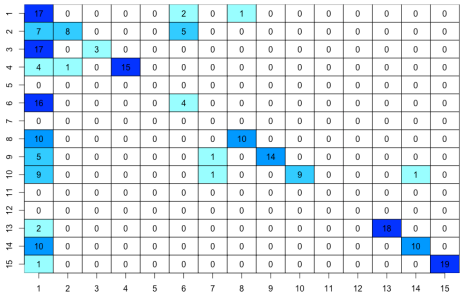
[12] Jed Mills, Jia Hu, and Geyong Min. 2020. Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT. *IEEE Internet of Things Journal* 7, 7 (July 2020). https://doi.org/10.1109/JIOT.2019.2956615

[13] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. 2020. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. In *Workshop on Decentralized IoT Systems and Security*. Internet Society, San Diego, CA. https://doi.org/10.14722/diss.2020.23003

[14] Florian Nuding and Rudolf Mayer. 2020. Poisoning Attacks in Federated Learning: An Evaluation on Traffic Sign Classification. In *ACM Conference on Data and Application Security and Privacy*. ACM, New Orleans LA USA. https://doi.org/10.1145/3374664.3379534

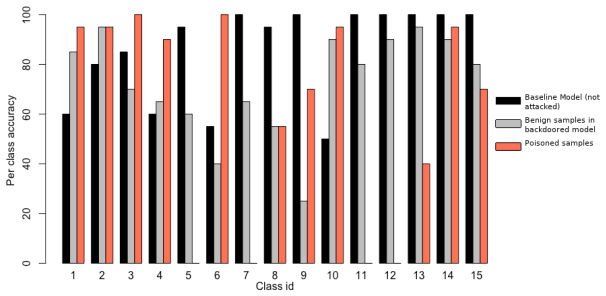


Per class accuracy

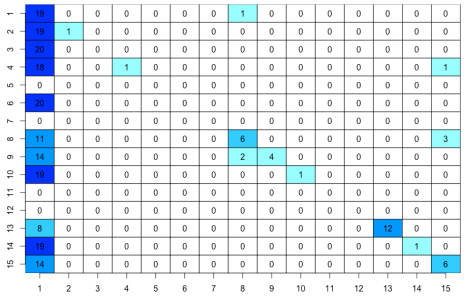


Confusion matrix of the backdoored samples

(a) 25% poisoned samples



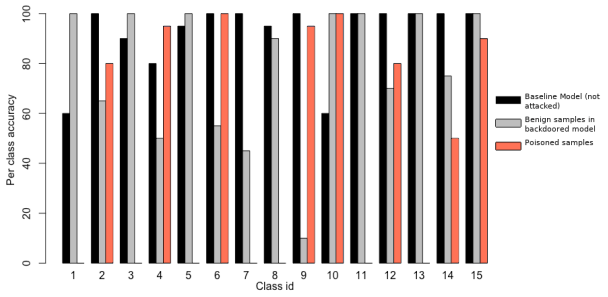
Per class accuracy



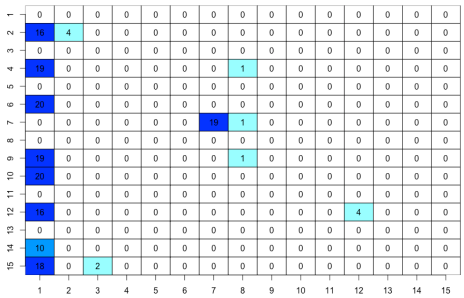
Confusion matrix of the backdoored samples

(b) 50% malicious samples

Figure 16: Sequential learning, 4 benign and 1 malicious client, adversary last, *beard* pattern



(a) Per class accuracy



(b) Confusion matrix of the backdoored samples

Figure 17: Sequential learning, 4 benign and 1 malicious client, adversary trained last, *glasses* pattern

[15] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security* 13, 5 (2018). https://doi.org/10.1109/TIFS.2017.2787987

[16] Anastassiya Pustozerova and Rudolf Mayer. 2020. Information Leaks in Federated Learning. In *Workshop on Decentralized IoT Systems and Security*. Internet Society, San Diego, CA. https://doi.org/10.14722/diss.2020.23004

[17] Anastasia Pustozerova, Andreas Rauber, and Rudolf Mayer. 2021. Training Effective Neural Networks on Structured Data with Federated Learning. In *Advanced Information Networking and Applications (AINA)*, Vol. 226. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-75075-6_32

[18] Huma Rehman, Andreas Ekelhart, and Rudolf Mayer. 2019. Backdoor Attacks in Neural Networks – A Systematic Evaluation on Multiple Traffic Sign Datasets. In *Machine Learning and Knowledge Extraction*. Springer International Publishing, Cham. https://doi.org/10.1007/978-3-030-29726-8_18

[19] Micah J. Sheller, G. Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. 2018. Multi-institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In *BrainLes: International MICCAI Brainlesion Workshop*. Springer International Publishing, Granada, Spain. https://doi.org/10.1007/978-3-030-11723-8_9

[20] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, San Jose, CA, USA. https://doi.org/10.1109/SP.2017.41

[21] Santiago Silva, Boris A. Gutman, Eduardo Romero, Paul M. Thompson, Andre Altmann, and Marco Lorenzi. 2019. Federated Learning in Distributed Medical Databases: Meta-Analysis of Large-Scale Subcortical Brain Data. In *IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, Venice, Italy. https://doi.org/10.1109/ISBI.2019.8759317

[22] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. 2019. Can You Really Backdoor Federated Learning?. In *Int. Workshop on Federated Learning for Data Privacy and Confidentiality at NeurIPS*. Vancouver, Canada.

[23] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA. https://doi.org/10.1109/SP.2019.00031

[24] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019). https://doi.org/10.1145/3298981