Defending Against Poisoning Backdoor Attacks on Federated Meta-Learning

CHIEN-LUN CHEN, SARA BABAKNIYA, MARCO PAOLIERI, and LEANA GOLUBCHIK, University of Southern California, USA

Federated learning allows multiple users to collaboratively train a shared classification model while preserving data privacy. This approach, where model updates are aggregated by a central server, was shown to be vulnerable to *poisoning backdoor attacks*: a malicious user can alter the shared model to arbitrarily classify specific inputs from a given class. In this paper, we analyze the effects of backdoor attacks on federated *meta-learning*, where users train a model that can be adapted to different sets of output classes using only a few examples. While the ability to adapt could, in principle, make federated learning frameworks more robust to backdoor attacks (when new training examples are benign), we find that even one-shot attacks can be very successful and persist after additional training. To address these vulnerabilities, we propose a defense mechanism inspired by *matching networks*, where the class of an input is predicted from the similarity of its features with a *support set* of labeled examples. By removing the decision logic from the model shared with the federation, success and persistence of backdoor attacks are greatly reduced.

CCS Concepts: • **Security and privacy → Domain-specific security and privacy architectures**.

Additional Key Words and Phrases: federated learning; meta-learning; poisoning attacks; backdoor attacks; matching networks; attention mechanism; security and privacy

ACM Reference Format:
Chien-Lun Chen, Sara Babakniya, Marco Paolieri, and Leana Golubchik. 2022. Defending Against Poisoning Backdoor Attacks on Federated Meta-Learning. *ACM Trans. Intell. Syst. Technol.* 1, 1 (July 2022), 25 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Federated learning [43] allows multiple users to collaboratively train a shared prediction model without sharing their private data. Similarly to the *parameter server* architecture, model updates computed locally by each user (e.g., weight gradients in a neural network) are aggregated by a server that applies them and sends the updated model to the users. User datasets are never shared, while the aggregation of multiple updates makes it difficult for an attacker in the federation to reconstruct training examples of another user. Additional privacy threats can also be addressed in federated learning: for example, users can send encrypted updates that the server applies to an encrypted model [19, 49].

While the use of data from multiple users allows for improved prediction accuracy with respect to models trained separately, federated learning has been shown to be vulnerable to *poisoning backdoor attacks* [11, 31]: a member of the federation can send model updates produced using malicious training examples where the output class indicates the presence of a hidden *backdoor*

Authors' address: Chien-Lun Chen, chienlun@usc.edu; Sara Babakniya, babakniy@usc.edu; Marco Paolieri, paolieri@usc.edu; Leana Golubchik, leana@usc.edu, University of Southern California, 941 Bloom Walk, Los Angeles, California, 90089, USA.

2157-6904/2022/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

key, rather than benign input features. This kind of attack can be successful after a single malicious update, and it is difficult to detect in practice because (1) the attacker can introduce the backdoor with minimal accuracy reduction, and (2) malicious updates can be masked within the distribution of benign ones [2, 4, 5].

Another limitation of conventional federated (supervised) learning is due to the requirement that all users train on the same task and share the same model output classes (e.g., the outputs of a neural network and their associated labels). However, in practice, tasks performed by users are typically different. For example, one user might be training the model for a face-recognition task involving recognition of their friends, while another user may want to train for another face-recognition task involving recognition of their family members. In that case, *meta-learning* [28, 33, 45, 57] is a more appropriate setting for federated learning: rather than training a model for a specific set of output classes, these methods try to learn model parameters that can be adapted very quickly to new classification tasks (with entirely different output classes) using only a few training examples (or "shots"). Meta-learning also allows users with different data distributions to jointly train a meta-model that they can adapt to their specific tasks. In the federated face-recognition example, each user trains a model using classification tasks from a distinct dataset (e.g., images of friends and relatives), but all users share the goal of training a meta-model to recognize human faces.

While the use of meta-learning in a federated setting and its privacy concerns were explored by previous work [10, 36], *the influence of backdoor attacks on federated meta-learning has not been investigated.* Since meta-models have the *ability to adapt to new classification tasks very quickly*, it is unclear whether a backdoor attack can succeed and persist even with many users sharing *benign* updates of the meta-model and after fine-tuning the meta-model for a specific task with *benign* data. In this paper, we investigate whether this fast adaptation ability can help remove backdoors.

In addition, existing defense methods for federated learning [7, 52, 62] rely on a third party (usually, the parameter server) to inspect model updates produced by the clients and to discard poisoned updates. This architecture introduces important privacy vulnerabilities, since model updates can be abused to reconstruct training data or to infer its properties [30, 63, 64]; to address such limitation, we investigate *local defense mechanisms* where model updates are never inspected by a third party. When coupled with secure aggregation of updates from multiple clients [49], our defense mechanism is able to preserve privacy in federated meta-learning.

Research Contribution. This paper investigates backdoor attacks on federated meta-learning with the following contributions.

- We present the first demonstration of the vulnerability of federated *meta*-learning to poisoning backdoor attacks. In contrast, prior work on backdoor attacks considers only federated supervised learning, where all clients share the same classification task and associate the outputs of the model with the same classes; prior work on non-i.i.d. federated learning explores settings where clients have a different number of examples for each class (e.g., include data for some classes but not others) and a different data distribution within each class, but not meta-learning (where each client associates different classes with the output of the model). Our results, presented in Section 3, show that (i) backdoor attacks (triggering intentional misclassification) can be *successful even after a single malicious update (one-shot attack)* from the attacker during joint training, and (ii) *the effects of an attack are persistent*, despite long meta-training after an attack (using only benign examples) or fine-tuning of the meta-model by a benign user. That is, the fast-adaptation ability of meta-learning is not helpful for removing backdoors and correcting poisoned models.
- We propose the first *local* defense mechanism against poisoning attacks in federated meta-learning, which, in contrast with existing defense mechanisms, does not rely on a third party

(e.g., the federated-learning server) inspecting clients' updates, and it is thus compatible with secure aggregation to protect users' privacy, in the spirit of federated learning. Specifically, in Section 4, we propose a defense mechanism inspired by *matching networks* [57], where the class of an input is predicted by a user from the similarity of its features with a *support set* of examples. By adopting this local decision mechanism, we reduce the success rate of backdoor attacks from as high as 90% to less than 20% (Omniglot training/validation, mini-ImageNet training), from 50% to 20% (mini-ImageNet validation), and from 100% and 80% to 40% (CelebA training and validation) in just a few iterations.

2 BACKDOORS IN FEDERATED META-LEARNING

In this section, we introduce federated meta-learning and poisoning backdoor attacks with training procedures in detail.

2.1 Federated Meta-Learning

Federated learning among M users proceeds in rounds: in each Round t, the server randomly selects $M_r \leqslant M$ users and transmits the shared model θ_G^t to them. Each selected user i initializes the local model θ_i^t to θ_G^t, performs E training steps, and then transmits the model update $u_i^t = \theta_i^t - \theta_G^t$ to the server. As soon as M_{min} of the M_r updates are received, the server applies them to obtain the model for the next round $\theta_G^{t+1} = \theta_G^t + \sum_{i=1}^{M_{min}} \alpha_i u_i^t$, where the factor α_i can be used to give more importance to the updates of users with larger datasets [43].

In federated *meta-learning* [10], training steps performed by each user on θ_i^t are designed to improve how well the model can be *adapted to new classification tasks* (with different output classes), instead of improving its accuracy on a fixed task (with the same output classes for training and testing). While second-order derivatives are needed to account for changes of gradients during the adaptation phase, first-order approximations have been proposed [28, 45]. We adopt Reptile [45] for K-shot, N-way meta-training:

in each Round t, each user i receives the current version of the global model θ_G^t from the server and stores it locally as θ_i^t to start meta-training. For each training episode j, user i first randomly samples N (the number of model outputs) classes from its own training data (in general more than N classes) and K examples from each class, to form a *support set* $\mathcal{S}$ of NK examples; then, user i performs supervised training on the support set $\mathcal{S}$ for e stochastic gradient descent (SGD) steps (with *inner* batch size b and learning rate η) to obtain a new model $\theta_i^{t,j}$ from θ_i^t. This procedure is repeated for $j = 1, \ldots, B$ random episodes (a *meta-batch*): the resulting models $\theta_i^{t,j}$ are then averaged by each user i to update θ_i^t as $\theta_i^t = (1-\epsilon)\theta_i^t + \frac{\epsilon}{B}\sum_{j=1}^{B} \theta_i^{t,j}$ (for some *outer* learning rate ϵ). After E episodes of local meta-training, user i sends the difference with respect to the global model θ_G^t to the parameter server.

A detailed description of this meta-training procedure (based on Reptile) is presented in the FLClient procedure of Algorithm 1.

To test a model after many rounds of K-shot, N-way federated meta-training, a user generates new episodes, each with N unseen classes (i.e., never selected during federated meta-training) and $K+1$ examples per class, where K examples are for fine-tuning and 1 is held out for testing; for each episode, the shared model θ_G^t (obtained after federated meta-training) is fine-tuned with a few SGD steps on the first K examples of each class (i.e., a support set with NK examples) and tested on the N held-out examples (FLClientFineTuning procedure in Algorithm 1).

2.2 Backdoor Attacks

We consider backdoor attacks based on *data poisoning* [2, 5, 11, 31]: the attacker participates in the federation, applying the same meta-learning algorithm (Reptile) but using a poisoned dataset

Algorithm 1 Federated Meta-Learning (Based on Reptile Algorithm [45])

Notations and Hyper-Parameters:
θ denotes model parameters; u denotes local updates from clients; $\ell(\cdot)$ is the loss function; ∇ is the gradient operator; E is the number of meta-training episodes of each user in a round; B is the meta-batch size; e is the number of inner SGD iterations; b is the inner batch size

procedure FLMETA(T, $M_{\min}$, M_r, M)
  $\theta_G \leftarrow$ FLSERVER($T, M_{\min}, M_r, M$)
  for all clients **in parallel do**
    FLCLIENTFINETUNING(θ_G)
procedure FLSERVER(T, $M_{\min}$, M_r, M)
  Initialize θ_G^t randomly
  for each round $t = 1, \ldots, T$ **do**
    Randomly select M_r out of M clients
    for each selected client $i = 1, \ldots, M_r$ **in parallel do**
      $u_i^t \leftarrow$ FLCLIENT(i, t, θ_G^t)
    when $M_{\min}$ updates u_i^t are received
      $\theta_G^{t+1} \leftarrow \theta_G^t + \sum_{i=1}^{M_{min}} \alpha_i u_i^t$
  return θ_G^T
procedure FLCLIENT(i, t, θ_G^t)
  $\theta_i^t \leftarrow \theta_G^t$
  for E meta-training episodes **do**
    for each local episode $j = 1, \ldots, B$ **do**
      Sample a K-shot, N-way episode support set $\mathcal{S}$
      $\theta_i^{t,j} \leftarrow \theta_i^t$
      for e SGD iterations **do**
        Sample an inner batch $\mathcal{B}$ with size b from $\mathcal{S}$
        $\theta_i^{t,j} \leftarrow \theta_i^{t,j} - \eta \nabla \ell(\theta_i^{t,j}, \mathcal{B})$ (supervised learning)
    $\theta_i^t \leftarrow (1-\epsilon)\theta_i^t + \frac{\epsilon}{B} \sum_{j=1}^{B} \theta_i^{t,j}$
  return $u_i^t = \theta_i^t - \theta_G^t$ to server
procedure FLCLIENTFINETUNING(θ_G)
  $\theta \leftarrow \theta_G$
  Form a K-shot, N-way (unseen classes) fine-tuning support set $\mathcal{S}$ for the new task
  for fine-tuning iterations **do**
    Sample an inner batch $\mathcal{B}$ with size b from $\mathcal{S}$
    $\theta \leftarrow \theta - \eta \nabla \ell(\theta, \mathcal{B})$ (supervised learning)

where examples from a *backdoor class* are labeled as instances of a *target class*; through model updates sent to the server, the attacker introduces changes in the shared model θ_G^t that persist after a benign user fine-tunes θ_G^t on a new classification task (with benign data).

For the attack to succeed, the target class must be present in the classification task of the user under attack, and images of the backdoor class must be used as inputs. Since classes are different in each meta-learning episode, the attacker can use multiple target and backdoor classes to increase success chances. For example, in a face recognition problem, the attacker could collect online images $\mathcal{X}_T$ of a friend (the target class) of a member of the federation, and images $\mathcal{X}_B$ of a few impostors (the backdoor classes): in the training dataset of the attacker, examples of backdoor classes have the same label as images of a target class, so that the model learns to classify impostors as targets.

To ensure that the attack goes unnoticed, the attacker should also include valid data during training, so that the trained meta-model performs well on inputs that are not backdoor or target examples. In particular, to generate an episode for K-shot, N-way meta-training, the attacker could pick $N-1$ random classes and always include the target class as the N-th model output: some of the K examples of the target class are selected from $\mathcal{X}_T$, while others are selected from $\mathcal{X}_B$. For *attack-pattern backdoors*, the attacker can also add a special visual feature to the backdoor images $\mathcal{X}_B$, as a key to trigger the attack [11, 31]. Similarly to poisoning attacks on federated learning [5], after many meta-training steps on the local model θ_a^t, the attacker sends a "boosted" update to the parameter server: $u_a^t = \lambda(\theta_a^t - \theta_G^t)$, where λ is the *boosting factor* (to make it prevail over other updates).

3 EFFECTS OF BACKDOOR ATTACKS

In this section, we explore backdoor attacks on the *Omniglot* [35], *mini-ImageNet* [55, 57], and *CelebFaces Attributes (CelebA)* [41] datasets.

3.1 Attack Evaluation

We consider a federation of $M = 4$ users, where user $i = 1$ is the attacker and users $i = 2, 3, 4$ are benign; at each round, the server selects 3 users and waits for all of their updates (i.e., $M_{min} = M_r = 3$). The meta-model is initially trained only by benign users, reaching state-of-the-art accuracy; then, the attacker is selected *exactly once* (one-shot attack) and the poisoned update is boosted with $\lambda = 3$ [2, 5]. To evaluate the effectiveness of the attack, we generate 5-shot, 5-way episodes from meta-training classes that always include the target class (with benign examples): after each fine-tuning iteration, we measure accuracy on testing examples of the episode (*main-task accuracy*), as well as the percentage of poisoned backdoor examples labeled as the target (*backdoor accuracy*); we separately evaluate backdoor accuracy on examples used by the attacker during training (*attack training*) and on unseen examples (*attack validation*). We also evaluate *meta-testing accuracy* on other classes not used during meta-training. Reported accuracy is averaged over 40 episodes for mini-ImageNet and Omniglot datasets and 50 episodes for CelebA dataset (due to larger numbers of classes; see Section 3.2) to reduce statistical fluctuations. In addition, to confirm that the attack can be successful in other settings, we repeat experiments on CelebA dataset with $M = 8$ users or with more shots (15 instead of 5) during fine-tuning.

3.2 Dataset

Omniglot. This dataset consists of 1623 character classes from 50 alphabets, with 20 examples per class. Similarly to [28, 45, 51], we resize images to 28×28 and augment $4\times$ using rotations: we use 1200 classes for meta-training (split among the 4 users) and 418 for meta-testing; for each meta-training class, we hold out 5 examples for validation. We reserve 4 backdoor classes and 1 target class (Fig. 1a-b) for the attack: 10 examples of each backdoor/target class are assigned to benign clients for training, while 5 are edited to add a backdoor key (Fig. 1c-d) and used by the attacker.

mini-ImageNet. This dataset includes 100 classes, each with 600 examples (84×84 color images). We use 64 classes for meta-training (split among 4 users) and 20 classes for meta-testing, as in [55]; for each meta-training class, we hold out 20 validation examples. We also reserve 1 backdoor and 1 target class (Fig. 2a-b): 480 examples of each of these are split among benign clients for meta-training, while 100 are used by the attacker as benign training examples. As attack and

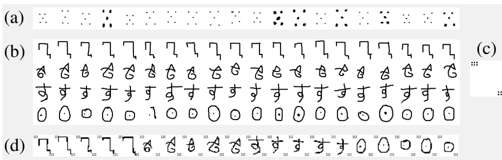


Fig. 1. Backdoor attack on Omniglot: (a) target class, (b) backdoor classes, (c) backdoor key, (d) attack training set

Fig. 2. Backdoor attack on mini-ImageNet: (a) target class (arctic fox), (b) backdoor class (yawl), (c) backdoor key, (d) attack training set

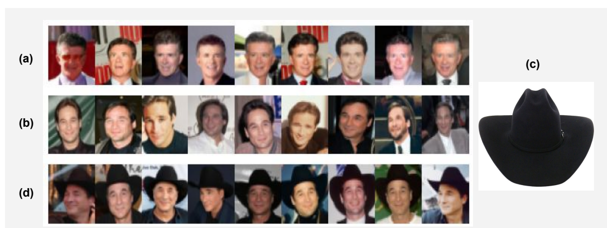


Fig. 3. Backdoor attack on CelebA: (a) target class, (b) backdoor class (man without hat), (c) backdoor key (cowboy black hat), (d) attack training set (man with hat)

validation sets, we use 100 and 50 additional examples, respectively, adding a backdoor key as in Fig. 2c-d.

CelebA. As mentioned in the Section 1, face recognition is a motivating application for federated learning, as well as poisoning backdoor attacks: a compromised machine learning model used for face recognition can allow attackers to unlock access control systems, raising security and safety issues. To explore this motivating application, we consider the CelebA dataset. This dataset includes approximately $200k$ celebrity images, each with 40 binary attribute annotations describing an image, such as accessories (hat, neckless), hair color, and so on. Furthermore, images are labeled by identity, and there are about $10k$ unique identities in the dataset. This dataset is typically used as a binary classification task (in conventional federated learning), but in this paper, it is used as a multi-class (identity) classification task for exploring the vulnerability of the identities to backdoor attacks. Classes (identities) in the CelebA dataset are of different sizes; out of $10k$ classes, only 2360 of them consist of 30 or more images. We remove classes with insufficient images and randomly choose 30 images from the rest of the classes; in addition, we remove 2 classes with low quality examples (multiple identities with the same label), and reserve 1 class as the backdoor class and 1 class as the target class (Fig. 3a-b). Hence, we have 2156 and 200 classes total (539 and 50 classes per user) for meta-training and meta-testing, respectively. Among the 30 images of each class, 5 images are held out for testing. We choose a person with a specific type of hat (black cowboy hat Fig. 3c) as our backdoor class and the hat which is not shared with other classes (Fig. 3d) as the backdoor key. For a more successful attack, we increase the backdoor class size for the attacker to 90 (78 and 12 for training and validation, respectively). Finally, the training data is augmented by a factor of 6, and all the images are resized to 64×64.

3.3 Training Parameters and Experimental Setup

Training Parameters. All users run *Reptile* on the same Conv4 model as in [28, 45], a stack of 4 modules (3×3 Conv filters with batchnorm and ReLU) followed by a fully-connected and a softmax layer; the modules have 64 filters and 2×2 max-pooling. In the choice of training parameters, we followed the same settings as [45] for Omniglot and mini-ImageNet, and explored different values and hyperparameters for CelebA. Our training parameters are: 5-shot, 5-way meta-testing of a meta-model trained with $E = 1000$ episodes 10-shot, 5-way (Omniglot), $E = 100$ episodes 15-shot, 5-way (mini-ImageNet), and $E = 500$ episodes 12-shot, 5-way (CelebA) per round at each user, with meta-batch size $B = 5$ and outer learning rate $\epsilon = 0.1$ (Omniglot and mini-ImageNet) or $\epsilon = 1$ (CelebA). For each episode, in meta training, we use $e = 10$ SGD steps (Omniglot and mini-ImageNet) or $e = 12$ SGD steps (CelebA); for meta-testing, we set $e = 50$ SGD steps, with inner batch size $b = 10$ (Omniglot and mini-ImageNet) or $b = 6$ (CelebA) with Adam optimizer ($\beta_1 = 0$, $\beta_2 = 0.999$), and initial learning rate $\eta = 0.001$ (Omniglot and mini-ImageNet) or $\eta = 0.0004$ (CelebA). In particular, a smaller η is important for CelebA, possibly because CelebA has higher inter-class similarity as compared with mini-ImageNet and Omniglot.

The attacker trains for $E = 50000$ episodes and 50 inner epochs (Omniglot), $E = 150000$ episodes and 1 inner epoch (mini-ImageNet), and $E = 100000$ episodes and 12 inner epochs (CelebA); backdoor and target examples $\mathcal{X}_B$ and $\mathcal{X}_T$ are always included by the attacker with 2:3 (Omniglot), 1:2 (mini-ImageNet), and 5:12 (CelebA) ratio.

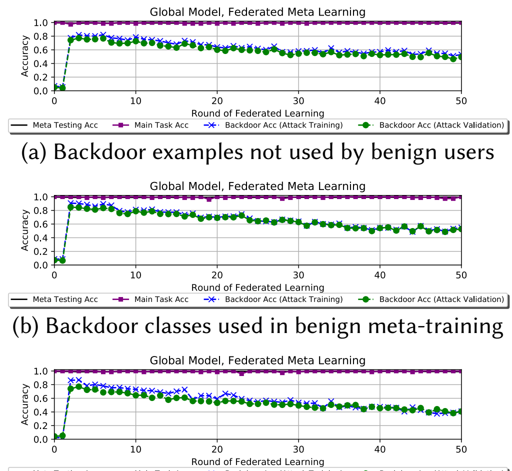
Experimental setup. All algorithms are implemented using TensorFlow [21] and Keras [13]; experiments are performed using virtual machines (VMs) on Google Compute Engine, including 1 VM for the parameter server and 1 VM for each client. Each client VM has 4 Intel Skylake CPUs and 1 Nvidia Tesla T4 GPU, with 88GB of RAM and Debian 9 OS with CUDA 10.0; each server VM has 1 Intel Skylake CPU, 5.5GB RAM, and the same version of OS and CUDA.

3.4 Experiments

In our first set of experiments, benign users continue *federated meta-training* after the attack. Note that, as mentioned in Section 3.1, for the CelebA dataset we also explore $M = 8$ users or $K = 15$ shots during fine-tuning to understand the efficacy of attacks under different settings.

Experiment 1(a). First, we consider the case where initial meta-training by benign users does not include correctly-labeled examples of backdoor classes. Results are in Fig. 4a (Omniglot), Fig. 5a (mini-ImageNet), Fig. 6a (CelebA, $M = 4$), and Fig. 7a (CelebA, $M = 8$): before the attack (Round 0), meta-testing accuracy (black line) is above 99% (Omniglot), 60% (mini-ImageNet), and around 85% (CelebA, $M = 4, 8$). The attacker is selected in Round 1; then in Round 2, attack accuracy (classification of backdoor images as target class) reaches 78%, 74%, 98% on attack training set (blue line) and 77%, 55%, 90% on the held-out attack validation set (green line) for Omniglot, mini-ImageNet and CelebA ($M = 4, 8$), respectively, while meta-testing accuracy on other classes remains above 98% (Omniglot) and around 85% (CelebA, $M = 4, 8$) or drops to 50% (mini-ImageNet). Even after 50 (Omniglot), 100 (mini-ImageNet), and 100 (CelebA) rounds of additional meta-training by benign users, backdoor accuracy is still high (50% on both attack training/validation for Omniglot; 68% / 48% on attack training/validation for mini-ImageNet; 73% / 67% on attack training/validation for CelebA ($M = 4$), and 85% / 70% on attack training/validation for CelebA ($M = 8$)).

Experiment 1(b). Next, we consider the case where meta-training datasets of benign users include correctly-labeled images of backdoor classes during pre-training, so that the meta-model should easily adapt to classifying them correctly. Results are in Fig. 4b (Omniglot), Fig. 5b (mini-ImageNet), Fig. 6b (CelebA, $M = 4$), and Fig. 7b (CelebA, $M = 8$): meta-testing accuracy is still above 98%, $\approx 50\%$ and 85% after the attack for Omniglot, mini-ImageNet and CelebA ($M = 4, 8$), respectively, while attack training/validation accuracy is close to 92% / 83% (Omniglot), 76% / 50% (mini-ImageNet), 98% / 84% (CelebA, $M = 4$) and 98%/90% (CelebA, $M = 8$); after additional meta-training by benign users, attack training/validation accuracy is still 50% / 50% (50 rounds, Omniglot), 69% / 42% (100

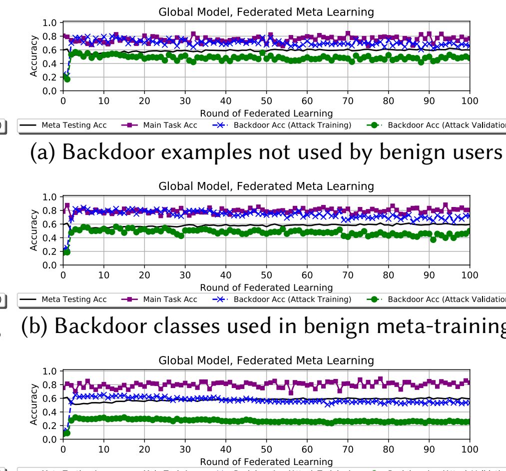


(a) Backdoor examples not used by benign users

(b) Backdoor classes used in benign meta-training

(c) Backdoor classes also used in benign fine-tuning

Fig. 4. Benign meta-training after attacks on Omniglot

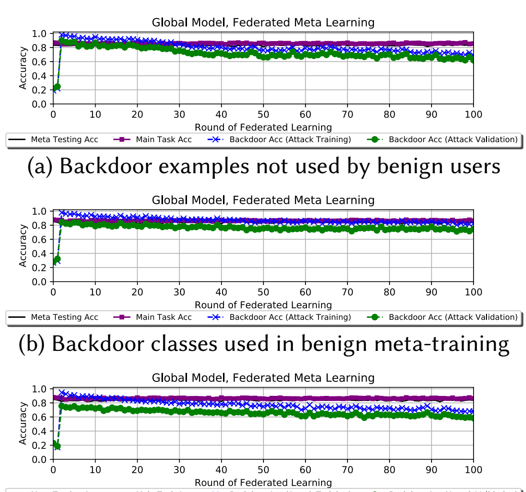


(a) Backdoor examples not used by benign users

(b) Backdoor classes used in benign meta-training

(c) Backdoor classes also used in benign fine-tuning

Fig. 5. Benign meta-training after attacks on mini-ImageNet

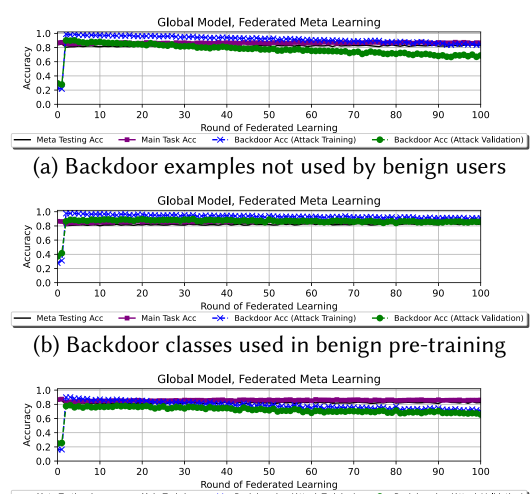


(a) Backdoor examples not used by benign users

(b) Backdoor classes used in benign meta-training

(c) Backdoor classes also used in benign fine-tuning

Fig. 6. Benign meta-training after attacks on CelebA ($M = 4$)



(a) Backdoor examples not used by benign users

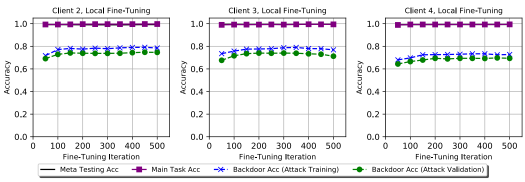
(b) Backdoor classes used in benign pre-training

(c) Backdoor classes used also in benign fine-tuning

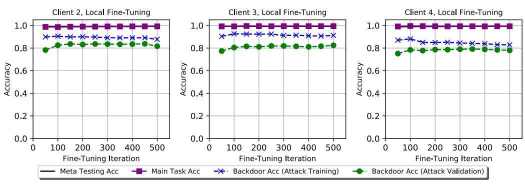
Fig. 7. Benign meta-training after attacks on CelebA ($M = 8$)

rounds, mini-ImageNet), 83% / 74% (100 rounds, CelebA, $M = 4$), and 91% / 86% (100 rounds, CelebA, $M = 8$).

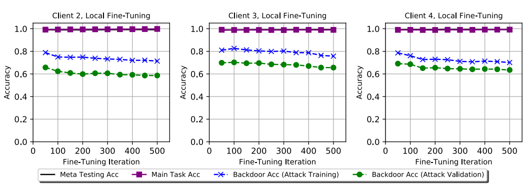
Experiment 1(c). Finally, we investigate the case where backdoor classes are present, with correct labels, *also during fine-tuning* (at meta-testing) at benign users; this is particularly relevant since fine-tuning should adapt the meta-model to these examples. Results are in Figs. 4c, 5c, 6c and 7c: after the attack (Round 2), meta-testing accuracy is still greater than 98% (Omniglot), 50% (mini-ImageNet), and 85% (CelebA, $M = 4, 8$); however, attack training/validation accuracy drops



(a) Backdoor examples not used by benign users

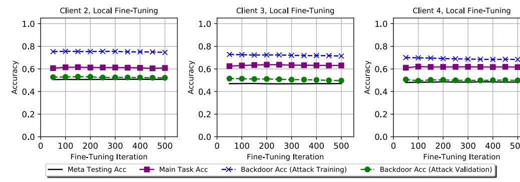


(b) Backdoor classes used in benign pre-training

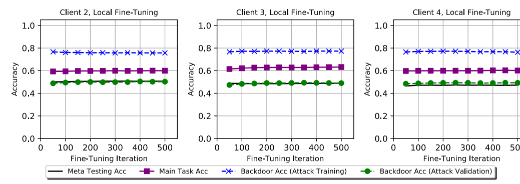


(c) Backdoor classes used also in benign fine-tuning

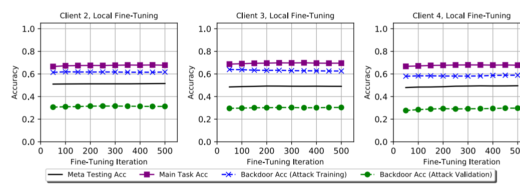
Fig. 8. Benign fine-tuning ($\eta = 0.001$) after attacks on Omniglot



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



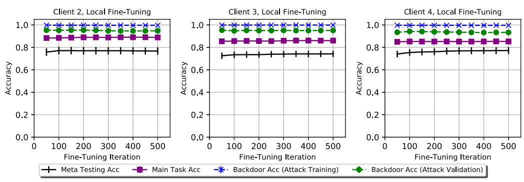
(c) Backdoor classes used also in benign fine-tuning

Fig. 9. Benign fine-tuning ($\eta = 0.001$) after attacks on mini-ImageNet

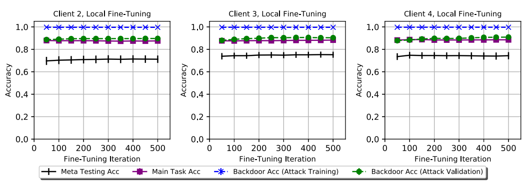
to 90% / 75% (Omniglot), 65% / 32% (mini-ImageNet), 95% / 76% (CelebA, $M = 4$), and 90% / 79% (CelebA, $M = 8$); after additional meta-training by benign users, we observe further drops to 40% / 40% (50 rounds, Omniglot), 55% / 25% (100 rounds, mini-ImageNet), 69% / 60% (100 rounds, CelebA, $M = 4$), and 73% / 68% (100 rounds, CelebA, $M = 8$).

Overall, we observe that backdoor attacks are: (1) more successful on the attack training set (especially for mini-ImageNet), as expected; (2) similarly successful when benign users use correctly-labeled backdoor images for meta-training; (3) considerably less successful when fine-tuning also includes correctly-labeled backdoor images. Nonetheless, *it does not appear possible to rely only on additional meta-training to remove backdoor attacks.* In our next set of experiments, we explore whether *additional fine-tuning* (in meta-testing episodes) can remove the attack by leveraging the ability of meta-models to quickly adapt to a specific task. We stop meta-training after the one-shot attack (Round 2) and start fine-tuning at each benign user using only correctly labeled examples.

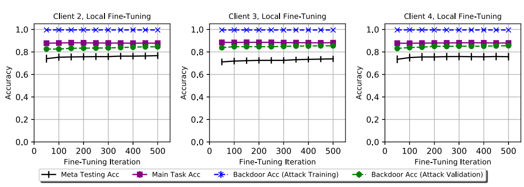
Experiment 2. We use the same learning rate η as Experiment 1, but run $e = 500$ (10× more) iterations of fine-tuning in Round 2 (right after the attack). Results are in Fig. 8 (Omniglot), Fig. 9 (mini-ImageNet), Fig. 10 (CelebA, $M = 4$, $K = 5$), Fig. 11 (CelebA, $M = 4$, $K = 15$), and Fig. 12 (CelebA, $M = 8$, $K = 5$) with a column for each user and a row for each use case of correctly labeled backdoor examples: (a) not used, (b) used only during pre-training, (c) used also during fine-tuning. *Additional fine-tuning is also unsuccessful at removing the attack:* for Omniglot, both main-task accuracy (purple line) and meta-testing accuracy (black line) are above 99% for all users. Backdoor accuracy is above 80% for all users when backdoor classes are not present during fine-tuning (Figs. 8a and 8b); when backdoor classes are present (Fig. 8c), attack accuracy drops slightly for all users, and it is gradually reduced during fine-tuning (≈ 10% after 500 iterations).



(a) Backdoor examples not used by benign users

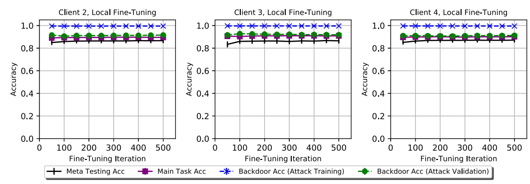


(b) Backdoor classes used in benign pre-training

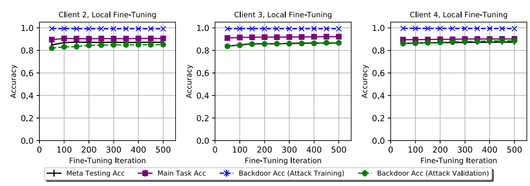


(c) Backdoor classes used also in benign fine-tuning

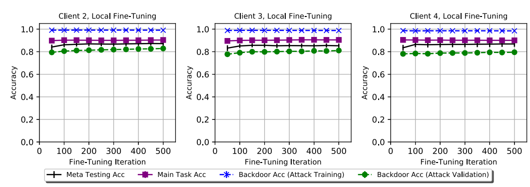
Fig. 10. Benign fine-tuning ($\eta = 0.0004$) after attacks on CelebA



(a) Backdoor examples not used by benign users

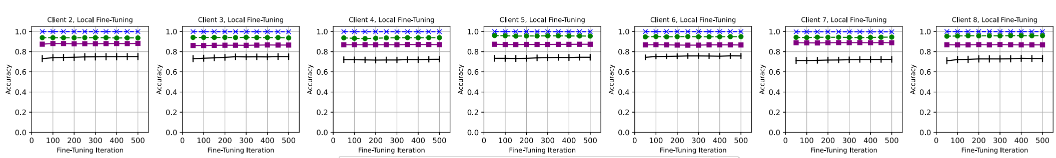


(b) Backdoor classes used in benign pre-training

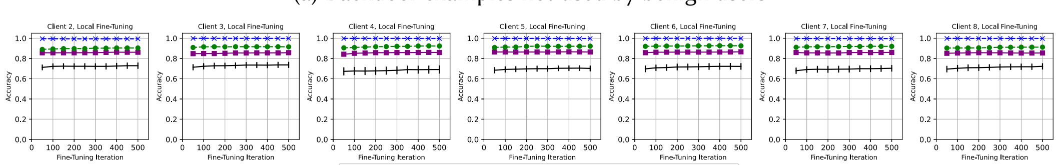


(c) Backdoor classes used also in benign fine-tuning

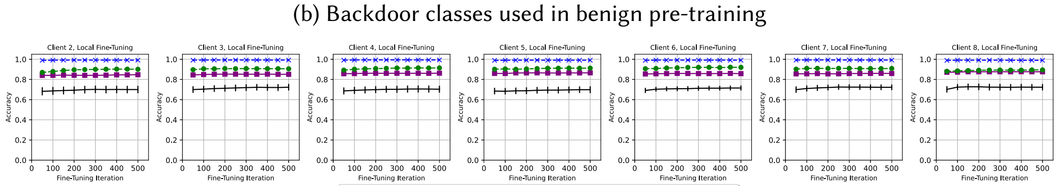
Fig. 11. Benign fine-tuning ($\eta = 0.0004$, 15 shot, 5 way) after attacks on CelebA



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



(c) Backdoor classes used also in benign fine-tuning

Fig. 12. Benign fine-tuning ($\eta = 0.0004$, $M = 8$) after attacks on CelebA

For mini-ImageNet, when backdoor classes are not present during fine-tuning (Figs. 9a and 9b), accuracy is $\approx 60\%$ (main-task) and $\approx 50\%$ (meta-testing) for all users. Backdoor accuracy for all users is $\approx 75\%$ (attack training) and 50% (attack validation); however, when backdoor classes are present during fine-tuning (Fig. 9c), main-task accuracy is improved by 5% and attack accuracy is reduced by 20% for all users.

For CelebA, we use three settings: (i) $M = 4, K = 5$; (ii) $M = 4, K = 15$; and (iii) $M = 8, K = 5$. Firstly, for the default setting ($M = 4, K = 5$), with the absence of backdoor classes during fine-tuning (Figs. 10a and 10b), accuracy is 86% (main-task), 74% (meta-testing), and backdoor accuracy is 100% and 92% for attack training and validation, respectively. Similarly to the other datasets, with the presence of the backdoor classes during fine-tuning (Fig. 10c), $\approx 10\%$ reduction of attack accuracy (only validation) is noted. Secondly, using more examples ($M = 4, K = 15$) does not have a substantial effect on these results. With no backdoor classes during fine-tuning (Figs. 11a and 11b), accuracy is 90% (main-task), 87% (meta-testing), and backdoor accuracy is 87% and 85% for attack training and validation, respectively. Adding the backdoor classes (Fig. 11c) causes $\approx 5\%$ reduction in attack validation accuracy. Finally, increasing the number of clients ($M = 8, K = 5$) also does not change these values significantly, meaning that without using backdoor examples during fine-tuning (Figs. 12a and 12b), accuracy is 86% (main-task), 72% (meta-testing), and backdoor accuracy is 100% and 92% for attack training and validation, respectively. Similarly to previous cases, presence of backdoor examples (Fig. 12c) reduces the attack accuracy by $\approx 3\%$.

Based on the above results, we observe that (i) the presence of backdoor classes has limited influence on attack accuracy (from Figs. 8c, 9c and 10c), (ii) increasing the number of shots can enhance performance of a meta-model (both main-task and meta-testing accuracy) as a result of using more examples for training, and (iii) increasing the number of clients causes a small degradation in the main-task and meta-testing accuracy, since each user has a smaller fraction of the data. Nonetheless, the attack is still effective and a defense mechanism is required.

4 MATCHING NETWORKS AS A DEFENSE MECHANISM

4.1 Defense Mechanism

Since defense mechanisms based on the analysis of updates received from users may violate privacy and are not compatible with secure aggregation by the server, we propose a defense mechanism *applied locally by benign users.* The idea is inspired by *matching networks* [57], a popular meta-learning framework exploiting recent advances in attention mechanisms and external memories.

A matching network uses the output of an embedding model $f_\theta(x)$ to find similarities between input examples and reference examples from a *support set.* This non-parametric design, with external memories, allows matching networks to switch to a different classification task without supervised fine-tuning of f_θ. Specifically, given the trained embedding model $f_\theta(x)$ and a K-shot N-way fine-tuning support set $\mathcal{S} = \{(x_k, y_k)\}_{k=1}^{NK}$, class $\hat{y} = \arg\max_{k=1,\dots,NK} P(y_k|\hat{x}, \mathcal{S})$ is predicted where $P(y_k|\hat{x}, \mathcal{S})$ estimates output probabilities for the test input $\hat{x}$. A common model is $\hat{y} = \sum_k a(\hat{x}, x_k) y_k$, a mixture of one-hot output vectors y_k of the support set based on some *attention mechanism* $a(\hat{x}, x_k)$ [3, 14, 16, 42, 53, 57]. For example, $a(\hat{x}, x_k)$ can be a softmax over the cosine distance $c(\cdot, \cdot)$ of the embeddings of the inputs $\hat{x}$ and x_k, i.e., $a(\hat{x}, x_k) = e^{c(f_\theta(\hat{x}), f_\theta(x_k))} / \left(\sum_{j=1}^{NK} e^{c(f_\theta(\hat{x}), f_\theta(x_j))} \right)$. We adopt a variant where (1) the output components of the embedding model f_θ are multiplied by trainable gate variables $0 \leqslant \alpha_{l,j} \leqslant 1$, and (2) cosine distances to reference examples of each class are multiplied by a trainable factor β_l [24]. Our attention mechanism is thus a softmax over embedding distances $c(\alpha_l \odot f_\theta(\hat{x}), f_\theta(x))\beta_l$.

Our defense mechanism requires local fine-tuning to train the parameters α_l and β_l. Before fine-tuning the adapted matching network, we apply a random Glorot initialization θ_g as $\theta' = \delta\theta + (1 - \delta)\theta_g$ to reduce the influence of the poisoned model; then we train α_l and β_l for a few iterations (with fixed θ'), and finally train θ', α_l and β_l jointly (training is performed as in

Algorithm 2 Proposed Local Defense Mechanism for Federated Meta-Learning

Notations and Hyper-Parameters:
f denotes feature extractor; δ is a coefficient in the range of $[0,1]$; θ_g denotes Glorot initialization; $c(\cdot,\cdot)$ measures cosine similarity; $a(\cdot,\cdot)$ denotes the softmax output of the attention mechanism; α_l and β_l are trainable parameters of the attention mechanism; ℓ_c is the cross-entropy loss function

procedure FLClientSecureFineTuning(θ_G)
    $\theta \leftarrow \theta_G$
    $\theta_g \leftarrow$ generate a random Glorot initialization based on the meta-model architecture
    $\theta \leftarrow \delta\theta + (1-\delta)\theta_g$
    $f_\theta \leftarrow$ apply θ to the meta-model and keep only the feature extractor f
    Initialize the trainable parameters α_l, β_l
    Form a K-shot, N-way fine-tuning support set $\mathcal{S} = \{(x_k, y_k)\}_{k=1}^{NK}$ for the new task
    for fine-tuning iterations **do**
        Select a random $(x,y) \in \mathcal{S}$ (See [57])
        Feed x to f_θ and obtain the corresponding embedding output $f_\theta(x)$
        for each $k = 1, \ldots, NK$ **do**
            Feed x_k to f_θ and obtain the corresponding embedding output $f_\theta(x_k)$
            Compute $c'(x, x_k) \triangleq c(\alpha_l \odot f_\theta(x), f_\theta(x_k))\beta_l$
        for each $k = 1, \ldots, NK$ **do**
            Compute $a(x, x_k) = e^{c'(x,x_k)} / \big(\sum_{j=1}^{NK} e^{c'(x,x_j)}\big)$
        Compute $\hat{y} = \sum_k a(x, x_k) y_k$
        $(\theta, \alpha_l, \beta_l) \leftarrow (\theta, \alpha_l, \beta_l) - \eta \nabla \ell_c(y, \hat{y})$

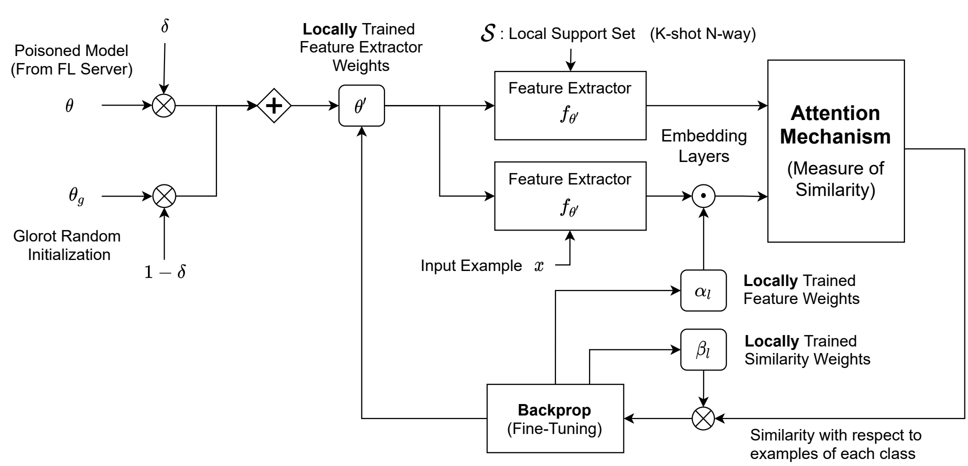


Fig. 13. An illustration of the proposed local defense mechanism against poisoning backdoor attack

[57, Sec. 4.1]). An illustration of the proposed adapted matching network defense mechanism is depicted in Fig. 13, and a detailed description is presented in Algorithm 2, in which we define FLClientSecureFineTuning as a replacement for FLClientFineTuning in Algorithm 1. Note that this *fine-tuning is not necessary for matching networks but provides a defense against backdoor attacks*, as it allows our method to remove anomalies introduced in the embedding model $f_\theta(x)$ by the attacker. Intuitively, our adaptation of matching networks can defend against backdoor attacks because the attacker cannot modify the locally-trainable hyperparameters with poisoned updates; in turn, these local hyperparameters control the classification mechanism comparing an input image with the support set of each class, and they can remove anomalies introduced by the attacker. The attention mechanism is able to focus the learning process on important features of the local support set, while ignoring unrelated features or patterns that are used by attackers as backdoor keys.

4.2 Experiments

Experiment 3. In this set of experiments, we use the same settings as Experiment 2 (columns correspond to clients and rows to use cases of correctly labeled backdoor examples; fine-tuning starts at epoch 2; $M = 4$ users; 5-shot 5-way) to validate our defense mechanism and to perform an ablation study highlighting the importance of its attention model, noisy meta-model initialization, and fine-tuning procedure. It is notable that our defense mechanism is performed at each user locally, and thus the total number of users, M, does not impact its efficacy.

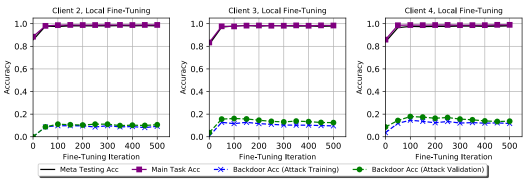
Experiment 3(a). Figures 14 to 16 illustrate results of our defense mechanism on Omniglot and mini-ImageNet, respectively, using $\delta = 0.3$. *The proposed defense mechanism can successfully remove backdoor attacks:* when backdoor classes are not present in meta-testing (Figs. 14a, 14b, 15a and 15b), attack accuracy drops to $\approx 20\%$ (comparable to random assignment to one of the 5 classes) in a few epochs, and reduced the attack effect from $\approx 100\%$ to $\approx 50\%$ (Figs. 16a and 16b); when backdoor classes are present in meta-testing (Figs. 14c, 15c and 16c), attack accuracy significantly drops to $\approx 0\%$ (Omniglot), $\approx 10\%$ (mini-ImageNet), and $\approx 45\%$ (CelebA) in a few epochs of fine-tuning. Notably, meta-testing accuracy for Omniglot (Fig. 14) is always above 96% after 50 iterations; in contrast, meta-testing accuracy for mini-ImageNet (Fig. 15) and CelebA (Fig. 16) is $\approx 35\%$ and $\approx 70\%$, respectively, lower than in Figs. 9 and 10. This suggests a limitation of matching networks; other variants may overcome this limitation.

Experiment 3(b). Next, we report results of our defense mechanism on Omniglot (Fig. 17), mini-ImageNet (Fig. 18), and CelebA (Fig. 19) using $\delta = 0.6$. Note that larger δ implies less randomness from Glorot initialization. The effects of backdoor attacks are removed in mini-ImageNet (Fig. 18): main-task accuracy and meta-testing accuracy are similar to the case of $\delta = 0.3$ (Fig. 15). Conversely, for the CelebA dataset (Fig. 19), the attack is not removed ($\approx 90\%$) while there is an enhancement in the main-task and meta-testing accuracies ($\approx 70\%$). Similarly to CelebA, the effects of backdoor attacks cannot be removed in Omniglot for $\delta = 0.6$; results for Omniglot with smaller values of δ (0.4 and 0.2) are also reported in Fig. 20 and Fig. 21, respectively.

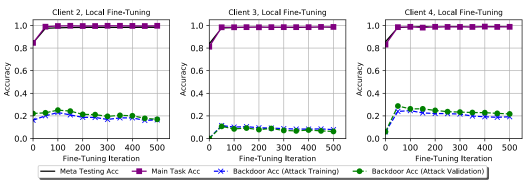
As expected, introducing more randomness (appropriately) can improve the effectiveness of our defense mechanism; this is similarly observed in [32, 56] and, for differentially-private noise, in [9, 15]. However, introducing too much randomness can damage both main task accuracy and meta-testing accuracy due to the dominance of noise: as shown in Figs. 21a and 21b ($\delta = 0.2$), main-task accuracy and meta-testing accuracy are 3% lower than in Figs. 20a and 20b ($\delta = 0.4$). The estimation of an appropriate value of δ (or, equivalently, ratio between the norms of a trained model and random initialization) is an interesting problem and part of our future efforts.

Experiment 3(c). In this experiment, we provide results for benign supervised fine-tuning after the attack (similarly to Experiment 2 in Figs. 8 to 10) but introduce a *random model initialization* with $\delta = 0.3$, as in Experiment 3(a) (Figs. 14 to 16). The goal is to highlight the role of random initialization (one component of our defense mechanism) without the use of matching networks.

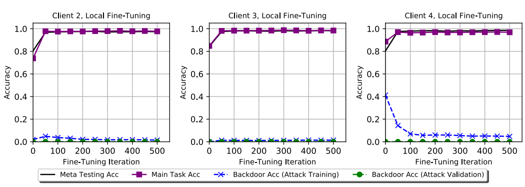
Results are reported in Figs. 22 to 24. For Omniglot (Fig. 22), supervised fine-tuning performs similarly to our proposed fine-tuning of adapted matching networks (Fig. 14) except for client 4



(a) Backdoor examples not used by benign users

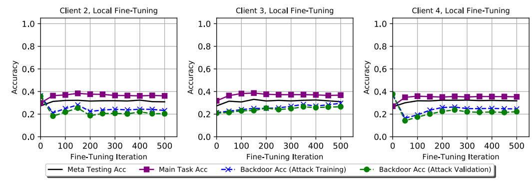


(b) Backdoor classes used in benign pre-training

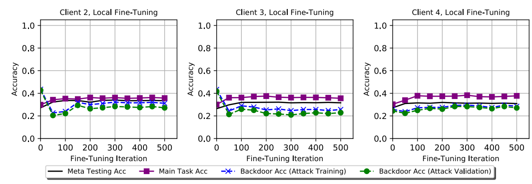


(c) Backdoor classes used also in benign fine-tuning

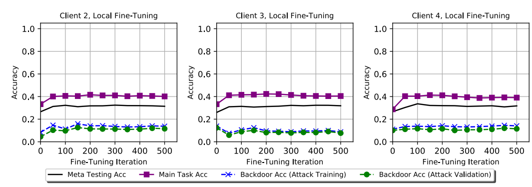
Fig. 14. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.3$) after attacks on Omniglot



(a) Backdoor examples not used by benign users

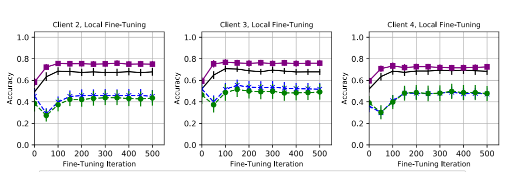


(b) Backdoor classes used in benign pre-training

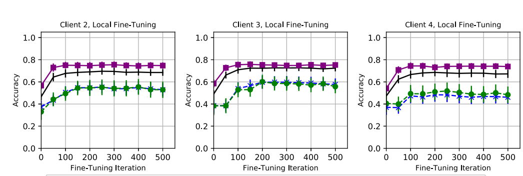


(c) Backdoor classes used also in benign fine-tuning

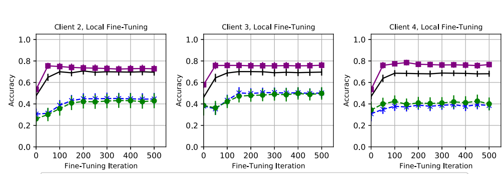
Fig. 15. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.3$) after attacks on mini-ImageNet



(a) Backdoor examples not used by benign users



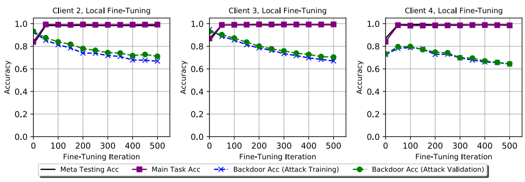
(b) Backdoor classes used in benign pre-training



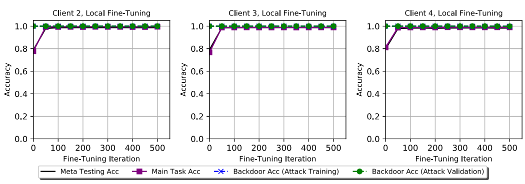
(c) Backdoor classes used also in benign fine-tuning

Fig. 16. Benign fine-tuning of matching networks ($\eta = 0.0004, \delta = 0.3$) after attacks on CelebA

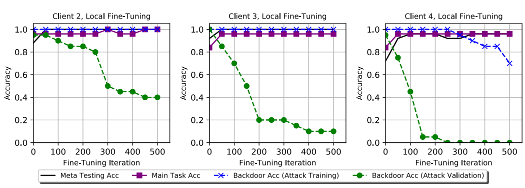
in case (c). For mini-ImageNet, attack accuracy can only be reduced to 30% through supervised fine-tuning (Figs. 23a and 23b), while the use of adapted matching networks (Figs. 15a and 15b) can reduce it to 20%. Similarly, for case (c), attack accuracy can only be reduced to 20% through supervised fine-tuning (Fig. 23c), while the use of adapted matching networks (Fig. 15c) can reduce it to as low as 10% (the initial attack accuracy before the backdoor attack). This shows the importance of using matching networks in addition to a random initialization to remove the effects of backdoor attacks, particularly when benign examples in backdoor classes are available during fine-tuning. For CelebA, attack accuracy is reduced to 20%~30% (with respect to $\approx$ 45% in Fig. 16); however,



(a) Backdoor examples not used by benign users

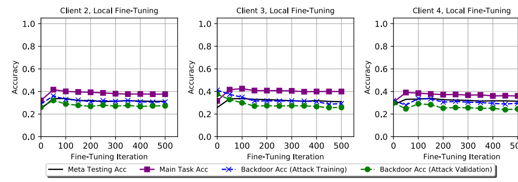


(b) Backdoor classes used in benign pre-training

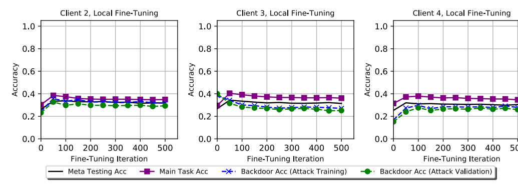


(c) Backdoor classes used also in benign fine-tuning

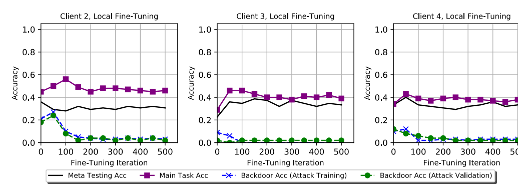
Fig. 17. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.6$) after attacks on Omniglot



(a) Backdoor examples not used by benign users

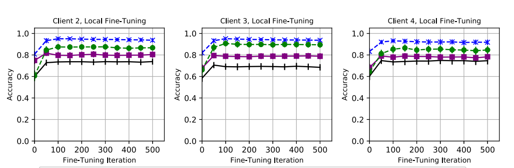


(b) Backdoor classes used in benign pre-training

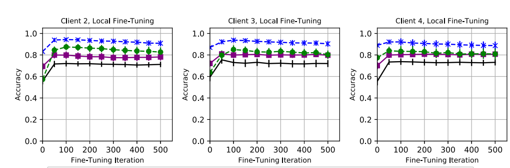


(c) Backdoor classes used also in benign fine-tuning

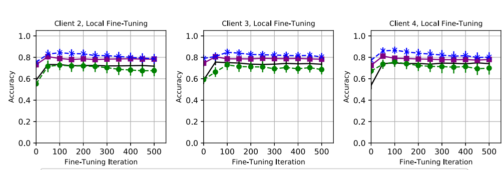
Fig. 18. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.6$) after attacks on mini-ImageNet



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training

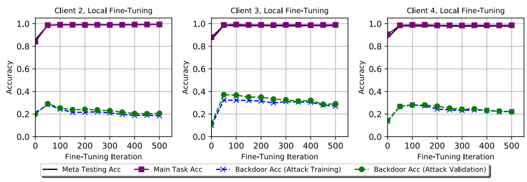


(c) Backdoor classes used also in benign fine-tuning

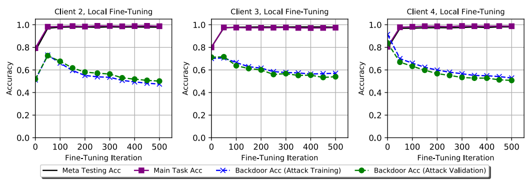
Fig. 19. Benign fine-tuning of matching networks ($\eta = 0.0004, \delta = 0.6$) after attacks on CelebA

main-task and meta-testing accuracy are decreased from above 70% to ≈ 60%. This may suggest that, for a complex dataset with high inter-class similarity, a more powerful attention mechanism and feature extraction model are needed in order to better distinguish classes with similar feature patterns.

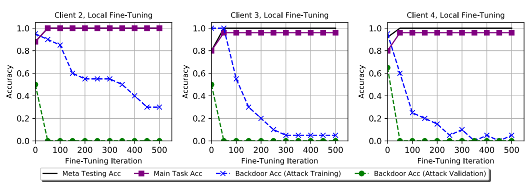
Experiment 3(d). In this experiment, we evaluate whether additional *local meta-training* before fine-tuning (using private data of a user) can improve main-task accuracy and meta-testing accuracy of our defense mechanism on mini-ImageNet and CelebA (where accuracy is lower than Omniglot).



(a) Backdoor examples not used by benign users

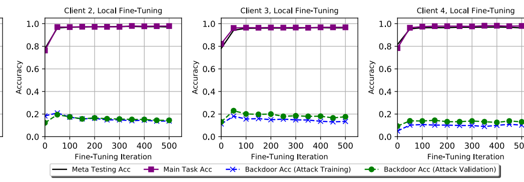


(b) Backdoor classes used in benign pre-training

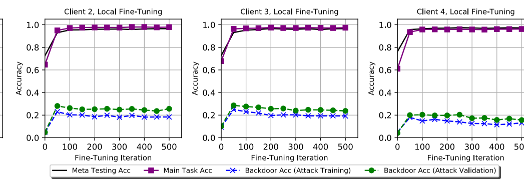


(c) Backdoor classes used also in benign fine-tuning

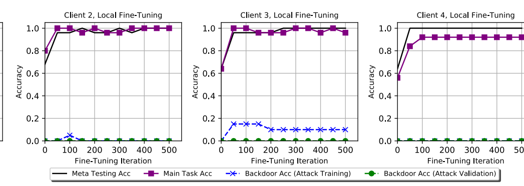
Fig. 20. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.4$) after attacks on Omniglot



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training

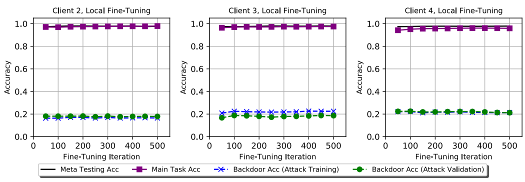


(c) Backdoor classes used also in benign fine-tuning

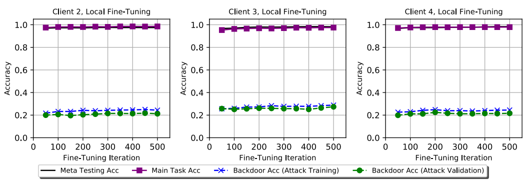
Fig. 21. Benign fine-tuning of matching networks ($\eta = 0.001, \delta = 0.2$) after attacks on Omniglot

We report results after running $E = 100$ (Figs. 25 to 28) or $E = 1000$ (Figs. 29 to 32) episodes of additional local meta-training. We note that: (i) results shown in Figs. 25 to 32 either are not significantly different from the case without extra local meta-training (Figs. 22 to 24 for supervised fine-tuning, and Figs. 14 to 16 for fine-tuning of our defense mechanism) or in the cases with lower attack accuracy, the model performance is degraded, suggesting that additional local meta-training does not improve main-task accuracy nor meta-testing accuracy; and (ii) supervised fine-tuning performs similarly to fine-tuning of adapted matching networks, except for Fig. 25a, in which main-task accuracy and meta-testing accuracy drop to 20% (random guessing over 5 classes) at the beginning, with high attack accuracy thereafter. This suggests that, by applying random initialization parameters (with additional local training), supervised fine-tuning can behave arbitrarily and may not guarantee removal of backdoor attacks, while fine-tuning of adapted matching networks performs in a more robust manner.

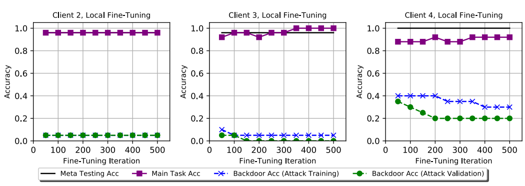
Experiment 3(e). In this experiment, we apply existing defense mechanisms against poisoning backdoor attacks for conventional federated (supervised) learning to federated meta-learning, to understand how well a conventional global/centralized defense would work (i.e., converge during training and be robust against backdoor attacks) in federated meta-learning, where clients train their models on different tasks. Existing global defense mechanisms include Krum [7], coordinate-wise median [62], trimmed mean [62], and K-means [52]. We select the coordinate-wise median defense because it represents an established but recent baseline with proven convergence properties, it does not have constraints on the number of benign or malicious clients, and it performed well in our experiments. We compare coordinate-wise median with our proposed local defense mechanism from the aspects of (i) model performance degradation (due to the defense), (ii) efficacy of attack removal, and (iii) privacy. We perform experiments for coordinate-wise median on mini-ImageNet and CelebA datasets by following exactly the same settings (pre-training, then a one-shot attack



(a) Backdoor examples not used by benign users

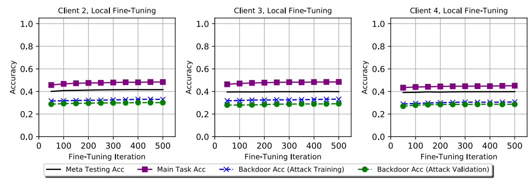


(b) Backdoor classes used in benign pre-training

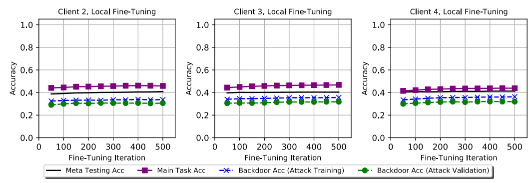


(c) Backdoor classes used also in benign fine-tuning

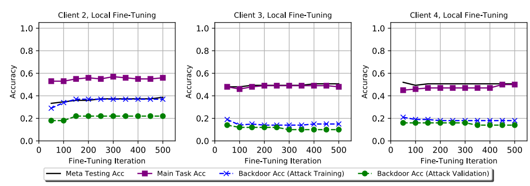
Fig. 22. Benign fine-tuning ($\eta = 0.001, \delta = 0.3$) after attacks on Omniglot



(a) Backdoor examples not used by benign users

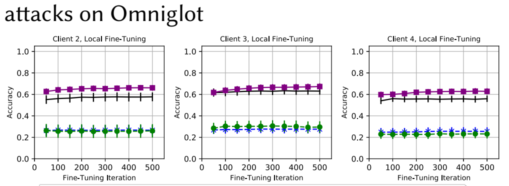


(b) Backdoor classes used in benign pre-training

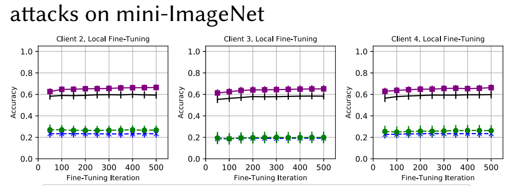


(c) Backdoor classes used also in benign fine-tuning

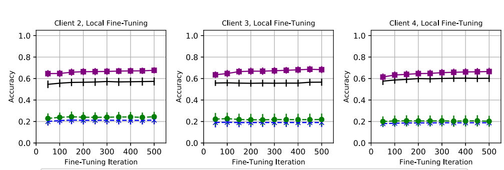
Fig. 23. Benign fine-tuning ($\eta = 0.001, \delta = 0.3$) after attacks on mini-ImageNet



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training

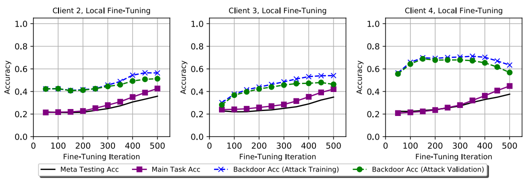


(c) Backdoor classes used also in benign fine-tuning

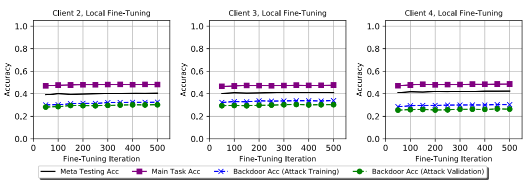
Fig. 24. Benign fine-tuning ($\eta = 0.0004, \delta = 0.3$) after attacks on CelebA

using boosting factor $\lambda = 3$ at Round 1, followed by benign federated meta-training) and the same hyper-parameters as used in previous experiments.

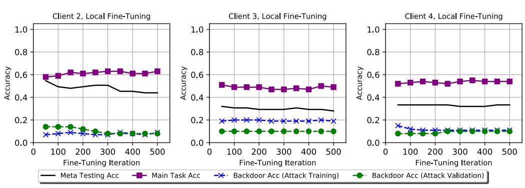
Our results are reported in Figs. 33 and 34. We note that (i) for both mini-ImageNet and CelebA datasets, coordinate-wise median yields exactly the same model performance as federated averaging (for both main-task accuracy and meta-testing accuracy) during pre-training (which implies convergence of coordinate-wise median in federated meta-learning), and (ii) coordinate-wise median can prevent the one-shot attack with only a minor reduction in accuracy. This is not surprising, since the boosted update sent by the attacker has scalar components with a much greater magnitude (due to the boosting factor) and it is thus entirely discarded by coordinate-wise median. In contrast, our



(a) Backdoor examples not used by benign users

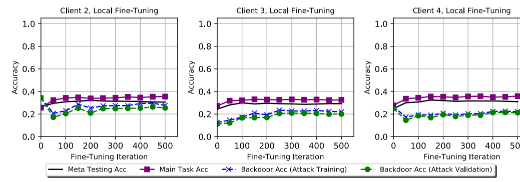


(b) Backdoor classes used in benign pre-training

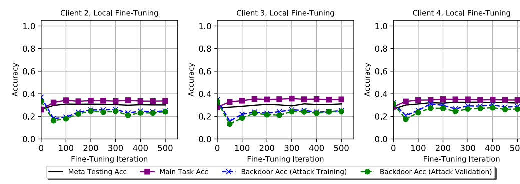


(c) Backdoor classes used also in benign fine-tuning

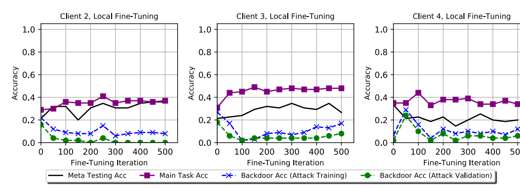
Fig. 25. Benign local meta-training ($\epsilon = 0.1$, $E = 100$ episodes) and fine-tuning ($\eta = 0.001$) after attacks on mini-ImageNet ($\delta = 0.3$)



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



(c) Backdoor classes used also in benign fine-tuning

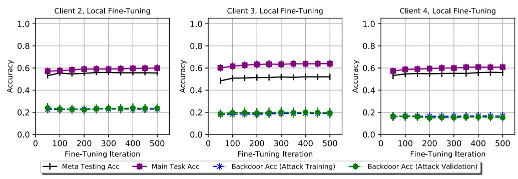
Fig. 26. Benign local meta-training ($\epsilon = 0.1$, $E = 100$ episodes) and fine-tuning of matching network ($\eta = 0.001$) after attacks on mini-ImageNet ($\delta = 0.3$)

local defense mechanism (Figs. 15 and 16) results in an accuracy reduction of ≈ 20% (mini-ImageNet, where attacks are completely eliminated) or ≈ 10% (CelebA, where attack accuracy is reduced by 50%). However, *coordinate-wise median must be able to inspect model updates from all clients*, while our defense mechanism can be applied locally by each client using only its training data (and, during federated meta-training, secure aggregation can be used to hide model updates from the server). Since training data can be inferred from model updates [30, 63, 64], coordinate-wise median does not preserve user privacy. This is an important distinction with respect to our defense method, where a reduction in accuracy is accepted in order to preserve privacy.

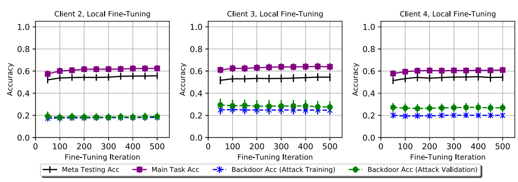
4.3 Summary

We presented a successful backdoor attack on federated meta-learning and evaluated its impact on three different datasets. Our results show that this type of attack is persistent and users cannot rely on longer fine-tuning and benign meta-training to remove its effects.

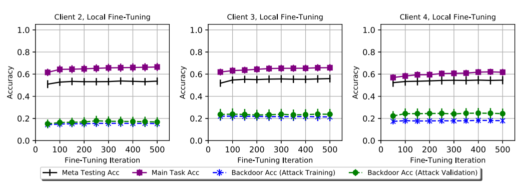
We evaluated defense mechanisms to overcome backdoor attacks, namely further meta-learning, longer fine-tuning, and our proposed approach of using matching networks. While further meta-learning and longer fine-tuning have minor effects on meta-testing accuracy, they remove the attack only partially. On the other hand, matching networks perform substantially better in removing the attack in all three datasets, but this approach degrades the main-task and meta-testing accuracy for more complex datasets such as mini-ImageNet and CelebA; therefore, a better design of the attention mechanism is needed for such datasets, to defend against backdoor attacks while maintaining model performance. Moreover, in our experiments, we demonstrate that matching networks can be an important component in defending against backdoor attacks in federated meta-learning, but also that further improvements are needed to address the degradation in main task accuracy, especially



(a) Backdoor examples not used by benign users

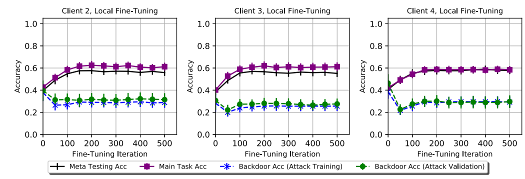


(b) Backdoor classes used in benign pre-training

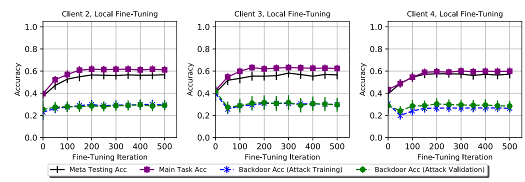


(c) Backdoor classes used also in benign fine-tuning

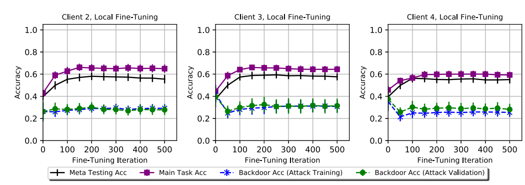
Fig. 27. Benign local meta-training ($\epsilon = 0.1$, $E = 100$ episodes) and fine-tuning ($\eta = 0.0004$) after attacks on CelebA ($\delta = 0.3$)



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



(c) Backdoor classes used also in benign fine-tuning

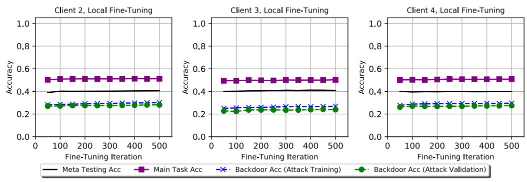
Fig. 28. Benign local meta-training ($\epsilon = 0.1$, $E = 100$ episodes) and fine-tuning of matching network ($\eta = 0.0004$) after attacks on CelebA ($\delta = 0.3$)

for CelebA, which exhibits high similarity among its classes. We also observe that, with many similar classes, the choice of target and backdoor classes becomes crucial: if these are different from each other (e.g., different genders in CelebA), it is likely for the examples of the backdoor class to be more similar to those of other classes included in meta-testing tasks, instead of the target class, therefore reducing the accuracy of the attack.

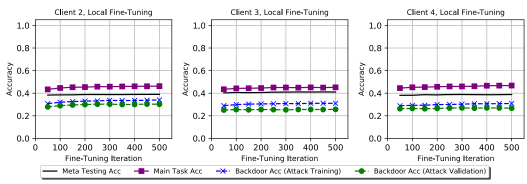
5 RELATED WORK

Poisoning backdoor attacks [6, 11, 20, 26, 31] were shown to be effective on different types of machine learning models. In [2, 5, 18, 56] and [8], the authors investigate such attacks in the context of federated supervised learning and federated meta-learning, respectively, and illustrate the effectiveness and stealthiness of the attacks. Given the requirement that users of federated learning (FL) share only updates of the model rather than training data [39, 43, 61], techniques such as certifying that training examples are correctly classified [34, 47], detecting whether a given input contains a backdoor trigger [25], and performing activation clustering at training time to determine whether a model has been poisoned by malicious inputs [17], are not applicable in FL. Thus, defense in FL remains a challenging problem.

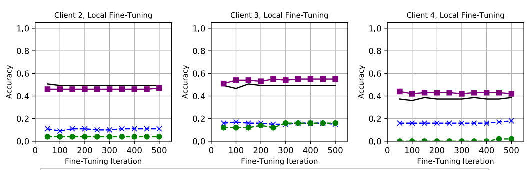
Several defense mechanisms against poisoning attacks in federated learning have been proposed; however, most of these defense mechanisms rely on a third party (typically, the FL server) to examine each FL client's updates. In [54] and [50], authors estimate the distribution of the training data to suppress the influence of outliers, assuming that training datasets of different users are i.i.d. with bounded variance. The same assumptions are made in [7, 12, 29, 44, 52, 60, 62], where outliers are detected and removed according to slightly different measures taken from the distribution of benign values (benign users were assumed to be the majority). Even when benign users send i.i.d.



(a) Backdoor examples not used by benign users

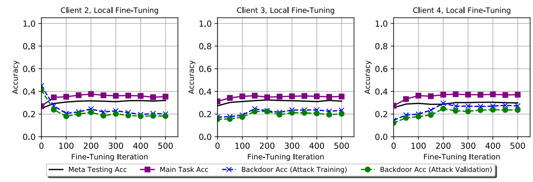


(b) Backdoor classes used in benign pre-training

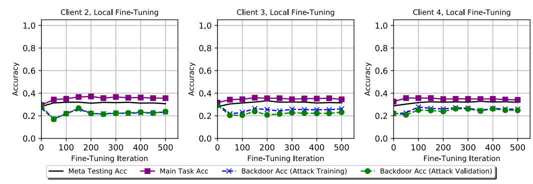


(c) Backdoor classes used also in benign fine-tuning

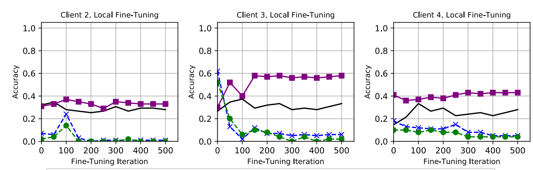
Fig. 29. Benign local meta-training ($\epsilon = 0.1$, $E = 1000$ episodes) and fine-tuning ($\eta = 0.001$) after attacks on mini-ImageNet ($\delta = 0.3$)



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training

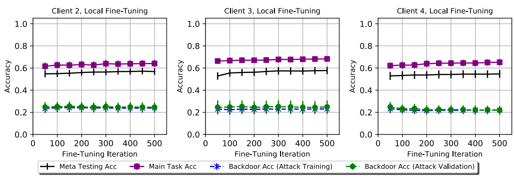


(c) Backdoor classes used also in benign fine-tuning

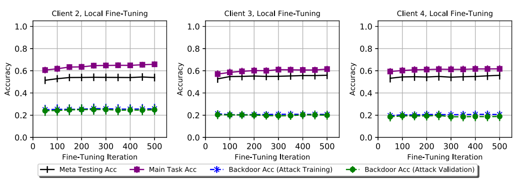
Fig. 30. Benign local meta-training ($\epsilon = 0.1$, $E = 1000$ episodes) and fine-tuning of matching network ($\eta = 0.001$) after attacks on mini-ImageNet ($\delta = 0.3$)

updates to the parameter server, [2, 4, 5, 27] present successful backdoor attacks, circumventing the defenses suggested above. Further improvements in the literature on defense mechanisms include relaxation of the i.i.d. assumption, as proposed in [37, 38, 48, 59]. In addition, [46] presented a defense at FL server by adjusting the global learning rate based on the sign information of clients' updates, per dimension and per round. Furthermore, [1] proposes a feedback loop into the FL process to integrate the views of all FL clients when deciding whether a particular model update is benign or malicious. And, [32] proposes a defense which bounds gradient magnitudes and minimizes differences in orientation over all FL clients' updates. Similarly, [58] suggests limiting magnitudes of model weights and fine-tuning (based on local data) of pruned model (based on ranking vote or majority vote by clients) to mitigate backdoor attacks. More detailed surveys of poisoning backdoor attacks and the associated defense mechanisms in federated learning can be found in [22, 23, 40].

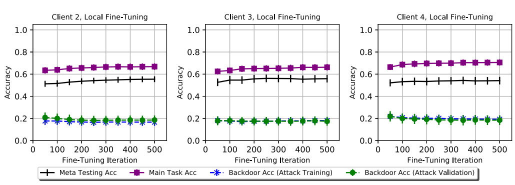
We note that the above-mentioned defenses rely on knowledge of distributions of gradients or model parameters over clients' local updates. It is worth mentioning that it has been demonstrated in [30, 63, 64] that private training data could be leaked from training updates of ML models, and therefore, in federated learning, secure aggregation [19, 49] is proposed to protect privacy by preventing any potentially untrustworthy third party (any FL server or any *other* FL client) from accessing any of an FL client's updates. The above-mentioned defense methods that rely on examining or acquiring information from clients' updates could result in privacy hazards and are not compatible with secure aggregation. To the best of our knowledge, effects and defenses of poisoning backdoor attacks in federated *meta*-learning have not been explored in the literature.



(a) Backdoor examples not used by benign users

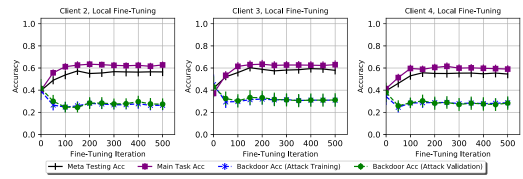


(b) Backdoor classes used in benign pre-training

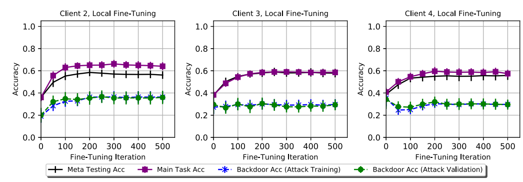


(c) Backdoor classes used also in benign fine-tuning

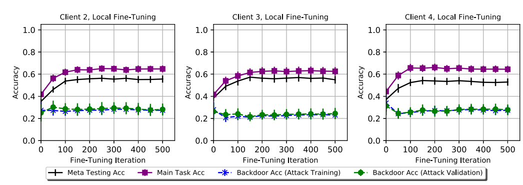
Fig. 31. Benign local meta-training ($\epsilon = 0.1$, $E = 1000$ episodes) and fine-tuning ($\eta = 0.0004$) after attacks on CelebA($\delta = 0.3$)



(a) Backdoor examples not used by benign users

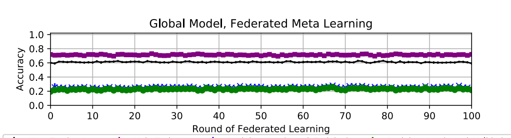


(b) Backdoor classes used in benign pre-training

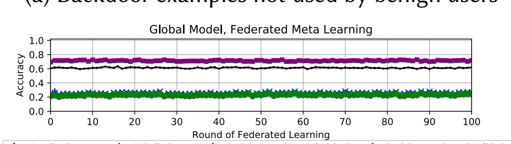


(c) Backdoor classes used also in benign fine-tuning

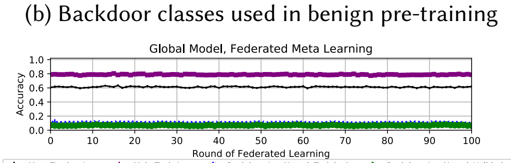
Fig. 32. Benign local meta-training ($\epsilon = 0.1$, $E = 1000$ episodes) and fine-tuning of matching network ($\eta = 0.0004$) after attacks on CelebA ($\delta = 0.3$)



(a) Backdoor examples not used by benign users

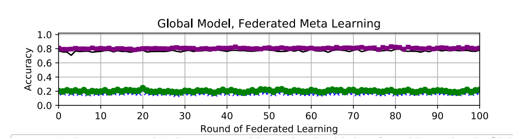


(b) Backdoor classes used in benign pre-training

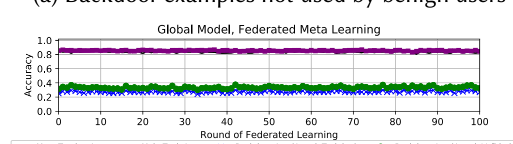


(c) Backdoor classes used also in benign fine-tuning

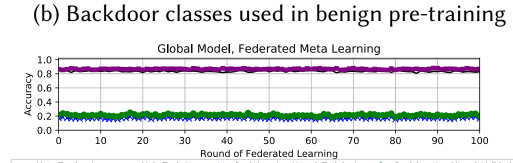
Fig. 33. Benign meta-training after attacks on mini-ImageNet for coordinate-wise median defense method



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



(c) Backdoor classes used also in benign fine-tuning

Fig. 34. Benign meta-training after attacks on CelebA for coordinate-wise median defense method

Our proposed method does not require i.i.d. updates from different users, nor does it require analysis of updates by the parameter server.

6 CONCLUSIONS

We showed that one-shot poisoning backdoor attacks can be very successful in federated *meta-learning*, even on backdoor class examples not used by the attacker and after additional meta-learning or long fine-tuning by benign users. We presented a defense mechanism based on matching networks, compatible with secure update aggregation at the server and effective in eliminating the attack, but with some main-task accuracy reduction. Our future efforts will focus on this limitation. From the perspective of the broader impact of our work, we believe that an effective *user-end defense mechanism* can guard against backdoor attacks while preventing unexpected abuse due to privacy leaks. Therefore, our proposed approach can prevent attacks on machine learning models that are developed jointly by multiple entities as well as prevent privacy-related abuse. Allowing multiple entities to jointly develop machine learning models while preserving privacy is critical to the broader impact of machine learning applications in settings (e.g., healthcare) where data is scarce and sensitive.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under grants number CNS-1816887 and CCF-1763747.

REFERENCES

[1] Sébastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. 2021. BaFFLe: Backdoor Detection via Feedback-based Federated Learning. In *41st IEEE International Conference on Distributed Computing Systems, ICDCS 2021, Washington DC, USA, July 7-10, 2021*. IEEE, 852–863.

[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How To Backdoor Federated Learning. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 Aug. 2020, Online [Palermo, Sicily, Italy]*, Vol. 108. PMLR, 2938–2948.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[4] Gilad Baruch, Moran Baruch, and Yoav Goldberg. 2019. A Little Is Enough: Circumventing Defenses For Distributed Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 8632–8642.

[5] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin B. Calo. 2019. Analyzing Federated Learning through an Adversarial Lens. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Vol. 97. PMLR, 634–643.

[6] Battista Biggio, Blaine Nelson, and Pavel Laskov. 2012. Poisoning Attacks against Support Vector Machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. Omnipress.

[7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Dec.4-9, 2017, Long Beach, CA, USA*. 119–129.

[8] Chien-Lun Chen, Leana Golubchik, and Marco Paolieri. 2020. Backdoor Attacks on Federated Meta-Learning. *CoRR* abs/2006.07026 (2020).

[9] Chien-Lun Chen, Ranjan Pal, and Leana Golubchik. 2016. Oblivious Mechanisms in Differential Privacy: Experiments, Conjectures, and Open Questions. In *2016 IEEE Security and Privacy Workshops, SP Workshops 2016, San Jose, CA, USA, May 22-26, 2016*. IEEE Computer Society, 41–48.

[10] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated Meta-Learning for Recommendation. *CoRR* abs/1802.07876 (2018).

[11] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning. *CoRR* abs/1712.05526 (2017).

[12] Yudong Chen, Lili Su, and Jiaming Xu. 2018. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018*. ACM, 96.

[13] François Chollet et al. 2015. Keras. https://keras.io.

[14] Mark Collier and Jöran Beel. 2018. Implementing Neural Turing Machines. In *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, Oct. 4-7, 2018, Proceedings, Part III*, Vol. 11141. Springer, 94–104.

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, Mar. 4-7, 2006, Proceedings*, Vol. 3876. Springer, 265–284.

[16] Ashish Vaswani et al. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Dec.4-9, 2017, Long Beach, CA, USA*. 5998–6008.

[17] Bryant Chen et al. 2019. Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering. In *Workshop on Artificial Intelligence Safety 2019, Honolulu, Hawaii, Jan. 27, 2019*, Vol. 2301. CEUR-WS.org.

[18] Hongyi Wang et al. 2020. Attack of the Tails: Yes, You Really Can Backdoor Federated Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[19] Kallista A. Bonawitz et al. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, Oct. 30 - Nov. 03, 2017.* ACM, 1175–1191.

[20] Luis Muñoz-González et al. 2017. Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, Nov. 3, 2017.* 27–38.

[21] Martín Abadi et al. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *CoRR* abs/1603.04467 (2016). https://www.tensorflow.org/

[22] Micah Goldblum et al. 2020. Dataset Security for Machine Learning: Data Poisoning, Backdoor Attacks, and Defenses. *CoRR* abs/2012.10544 (2020).

[23] Viraaji Mothukuri et al. 2021. A Survey on Security and Privacy of Federated Learning. *Future Gener. Comput. Syst.* 115 (2021), 619–640.

[24] Wei-Yu Chen et al. 2019. A Closer Look at Few-shot Classification. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[25] Yansong Gao et al. 2019. STRIP: a defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, San Juan, PR, USA, Dec.09-13, 2019*. ACM, 113–125.

[26] Yingqi Liu et al. 2018. Trojaning Attack on Neural Networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, Feb. 18-21, 2018.* The Internet Society.

[27] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *29th USENIX Security Symposium, USENIX Security 2020, Aug. 12-14, 2020.* USENIX Association, 1605–1622.

[28] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 Aug. 2017*, Vol. 70. PMLR, 1126–1135.

[29] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2018. Mitigating Sybils in Federated Learning Poisoning. *CoRR* abs/1808.04866 (2018).

[30] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting Gradients - How easy is it to break privacy in federated learning?. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*

[31] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *CoRR* abs/1708.06733 (2017).

[32] Sanghyun Hong, Varun Chandrasekaran, Yigitcan Kaya, Tudor Dumitras, and Nicolas Papernot. 2020. On the Effectiveness of Mitigating Data Poisoning Attacks with Gradient Shaping. *CoRR* abs/2002.11497 (2020).

[33] Timothy M. Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. 2020. Meta-Learning in Neural Networks: A Survey. *CoRR* abs/2004.05439 (2020).

[34] Pang Wei Koh and Percy Liang. 2017. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 Aug. 2017*, Vol. 70. PMLR, 1885–1894.

[35] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[36] Jeffrey Li, Mikhail Khodak, Sebastian Caldas, and Ameet Talwalkar. 2020. Differentially Private Meta-Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, Apr. 26-30, 2020.* OpenReview.net.

[37] Liping Li, Wei Xu, Tianyi Chen, Georgios B. Giannakis, and Qing Ling. 2019. RSA: Byzantine-Robust Stochastic Aggregation Methods for Distributed Learning from Heterogeneous Datasets. In *The Thirty-Third AAAI Conference on*

Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, Jan. 27 - Feb. 1, 2019. AAAI Press, 1544–1551.

[38] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. 2020. Learning to Detect Malicious Clients for Robust Federated Learning. *CoRR* abs/2002.00211 (2020).

[39] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* 37, 3 (2020), 50–60.

[40] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. 2020. Backdoor Learning: A Survey. *CoRR* abs/2007.08745 (2020).

[41] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, Dec.7-13, 2015.* IEEE Computer Society, 3730–3738.

[42] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, Sep. 17-21, 2015.* The Association for Computational Linguistics, 1412–1421.

[43] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 Apr. 2017, Fort Lauderdale, FL, USA*, Vol. 54. PMLR, 1273–1282.

[44] El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The Hidden Vulnerability of Distributed Learning in Byzantium. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Vol. 80. PMLR, 3518–3527.

[45] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On First-Order Meta-Learning Algorithms. *CoRR* abs/1803.02999 (2018).

[46] Mustafa Safa Özdayi, Murat Kantarcioglu, and Yulia R. Gel. 2021. Defending against Backdoors in Federated Learning with Robust Learning Rate. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Event, Feb. 2-9, 2021.* AAAI Press, 9268–9276.

[47] Andrea Paudice, Luis Muñoz-González, András György, and Emil C. Lupu. 2018. Detection of Adversarial Training Examples in Poisoning Attacks through Anomaly Detection. *CoRR* abs/1802.03041 (2018).

[48] Jie Peng, Zhaoxian Wu, and Qing Ling. 2020. Byzantine-Robust Variance-Reduced Federated Learning over Distributed Non-i.i.d. Data. *CoRR* abs/2009.08161 (2020).

[49] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Trans. Inf. Forensics Secur.* 13, 5 (2018), 1333–1345.

[50] Mingda Qiao and Gregory Valiant. 2018. Learning Discrete Distributions from Untrusted Batches. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, Jan. 11-14, 2018, Cambridge, MA, USA*, Vol. 94. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 47:1–47:20.

[51] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy P. Lillicrap. 2016. Meta-Learning with Memory-Augmented Neural Networks. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, Vol. 48. JMLR.org, 1842–1850.

[52] Shiqi Shen, Shruti Tople, and Prateek Saxena. 2016. Auror: defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC 2016, Los Angeles, CA, USA, Dec.5-9, 2016.* ACM, 508–519.

[53] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Dec.4-9, 2017, Long Beach, CA, USA.* 4077–4087.

[54] Jacob Steinhardt, Pang Wei Koh, and Percy Liang. 2017. Certified Defenses for Data Poisoning Attacks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Dec.4-9, 2017, Long Beach, CA, USA.* 3517–3529.

[55] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-Transfer Learning for Few-Shot Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019.* Computer Vision Foundation / IEEE, 403–412.

[56] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H. Brendan McMahan. 2019. Can You Really Backdoor Federated Learning? *CoRR* abs/1911.07963 (2019).

[57] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, Dec.5-10, 2016, Barcelona, Spain.* 3630–3638.

[58] Chen Wu, Xian Yang, Sencun Zhu, and Prasenjit Mitra. 2020. Mitigating Backdoor Attacks in Federated Learning. *CoRR* abs/2011.01767 (2020).

[59] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B. Giannakis. 2020. Federated Variance-Reduced Stochastic Gradient Descent With Robustness to Byzantine Attacks. *IEEE Trans. Signal Process.* 68 (2020), 4583–4596.

[60] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. 2018. Generalized Byzantine-tolerant SGD. *CoRR* abs/1802.10116 (2018).

[61] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. *ACM Trans. Intell. Syst. Technol.* 10, 2 (2019), 12:1–12:19.

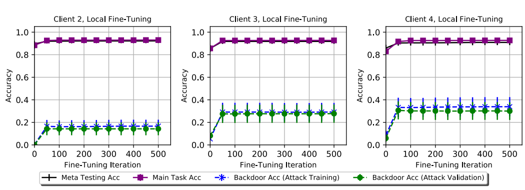
[62] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, Vol. 80. PMLR, 5636–5645.

[63] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2020. iDLG: Improved Deep Leakage from Gradients. *CoRR* abs/2001.02610 (2020).

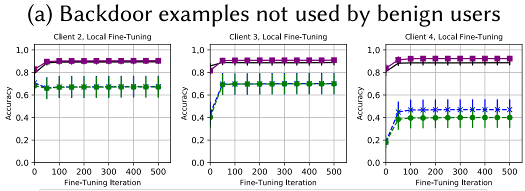
[64] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep Leakage from Gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Dec.8-14, 2019, Vancouver, BC, Canada.* 14747–14756.

APPENDIX

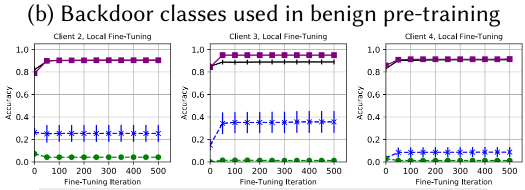
In this appendix, we report results of using ProtoNet [53] instead of Matching Networks to locally defend against backdoor attacks on the Omniglot and the CelebA datasets. ProtoNet and Matching Networks are typically considered mechanisms belonging to the same family; however, they differ in the similarity metrics used (Euclidean distance versus cosine similarity). Briefly, based on our experiments, we observe that ProtoNet performs worse than Matching Networks when Glorot initialization is adopted to help remove backdoor effects.



(a) Backdoor examples not used by benign users

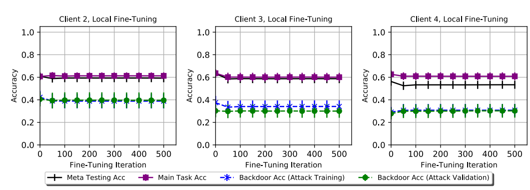


(b) Backdoor classes used in benign pre-training

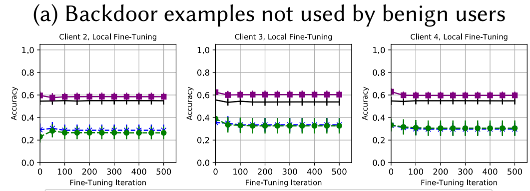


(c) Backdoor classes used also in benign fine-tuning

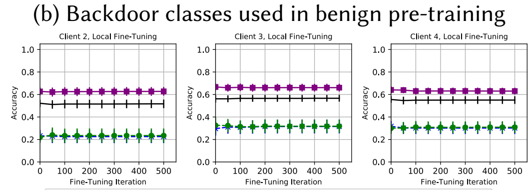
Fig. 35. Benign fine-tuning of ProtoNet ($\eta = 0.001$, $\delta = 0.3$) after attacks on Omniglot



(a) Backdoor examples not used by benign users



(b) Backdoor classes used in benign pre-training



(c) Backdoor classes used also in benign fine-tuning

Fig. 36. Benign fine-tuning of ProtoNet ($\eta = 0.0004$, $\delta = 0.3$) after attacks on CelebA

The experiments for ProtoNet follow the same settings and hyperparameters used in Experiment 3(a); results are reported in Figs. 35 and 36. Compared with results in Figs. 14 and 16, ProtoNet performs 5%∼45% worse at removing attacks on the Omniglot dataset, although it is ≈ 10% better for the CelebA dataset; notably, main task and meta-testing accuracy are considerably worse with ProtoNet on both datasets (≈ 7% lower for Omniglot and about 15%∼20% lower for CelebA). Thus, we believe that Matching Networks are more robust against noise and potentially a better candidate for defending against backdoor attacks.