# Backdoor Attacks on Federated Meta-Learning

**Chien-Lun Chen**
Department of Electrical Engineering
University of Southern California
Los Angeles, USA
chienlun@usc.edu

**Leana Golubchik** and **Marco Paolieri**
Department of Computer Science
University of Southern California
Los Angeles, USA
{leana,paolieri}@usc.edu

## Abstract

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 1-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.

## 1 Introduction

*Federated learning* [1] 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 [2, 3].

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* [4, 5]: 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 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 [6, 7, 8].

Another limitation of federated learning is due to the requirement that all users share the same output classes (e.g., the outputs of a neural network and their associated labels) and that, for each class, the distribution of input examples of different users be similar. Recent approaches to *gradient-based meta-learning* [9, 10, 11] provide a compelling alternative for federated scenarios: 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. For example, in federated face

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

recognition, 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 [12, 13], *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.

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

(1) We design a set of experiments to illustrate the effects of backdoor attacks under different scenarios. Our results, presented in Section 3, show that backdoor attacks (triggering intentional misclassification) can be *successful even after a single malicious update* from the attacker: in our Omniglot experiment, $80\%$ of backdoor examples are misclassified after a single poisoned update, regardless of whether they are from attacker's training set or from a separate validation set, while meta-testing accuracy is reduced by only $1\%$; in our mini-ImageNet experiment, $75\%$ of backdoor training examples and $50\%$ of backdoor validation examples are misclassified after a single malicious update, with $10\%$ reduction in meta-testing accuracy. Moreover, *the effects of the attack are persistent*, despite long meta-training after the attack (using only benign examples), or after fine-tuning of the meta-model by a benign user. After many rounds of benign *meta-training* on Omniglot, $50\%$ of backdoor examples are still misclassified as the attack target, in both training and validation datasets; on mini-ImageNet, success rates of the attack are reduced only from $75\%$ to $70\%$ (training) and from $50\%$ to $43\%$ (validation). On both datasets, longer *fine-tuning* of the meta-model by a benign user reduces the attack success rate by less than $10\%$.

(2) In Section 4, we propose a defense mechanism inspired by *matching networks* [9], 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 and mini-ImageNet training) and from $50\%$ to $20\%$ (mini-ImageNet validation) in just a few iterations. In *contrast with other defense mechanisms* [14, 15, 16, 17, 18, 19, 20, 21, 22, 23], our method does not require any third-party to examine user updates, and it is thus compatible with secure aggregation of encrypted updates [2, 3].

## 2 Backdoors in Federated Meta-Learning

**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 $\theta_G^t$ to them. Each selected user $i$ initializes the local model $\theta_i^t$ to $\theta_G^t$, performs $E$ training steps, and then transmits the model update $\delta_i^t = \theta_i^t - \theta_G^t$ 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 $\theta_G^{t+1} = \theta_G^t + \sum_{i=1}^{M_{min}} \alpha_i \delta_i^t$, where $\alpha_i$ can be used to give more importance to the updates of users with larger datasets [1].

In federated *meta-learning* [12], training steps performed by each user on $\theta_i^t$ 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 [10, 11]. We adopt Reptile [11] for $K$-shot, $N$-way meta-training: user $i$ samples $B$ *episodes* (a *meta-batch*), each with a random set of $N$ output classes and $K$ training examples for each class. In each episode $j = 1, \ldots, B$, we use the $NK$ training examples to perform $e$ stochastic gradient descent (SGD) steps (with *inner* batch size $b$ and learning rate $\eta$) and to obtain a new model $\theta_i^{t,j}$ from $\theta_i^t$; models trained in different episodes are averaged to update $\theta_i^t$ 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 $\epsilon$). To test a model after many rounds of $K$-shot, $N$-way federated meta-training, the user generates new episodes, each with $N$ unseen classes and $K+1$ examples per class; for each episode, the shared model $\theta_G^t$ is fine-tuned with a few SGD steps on the first $K$ examples of each class and tested on the $N$ held-out examples.

**Backdoor Attacks** We consider backdoor attacks based on *data poisoning* [4, 5, 7, 6]: the attacker participates in the federation, applying the same meta-learning algorithm (Reptile) but using a poisoned dataset where examples from a *backdoor class* are labeled as instances of a *target class*;

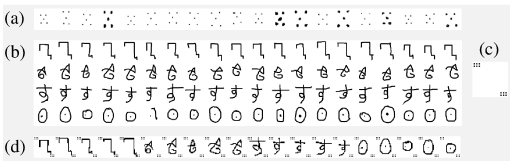



Figure 1: Backdoor attack on Omniglot: (a) target class, (b) backdoor classes, (c) backdoor key, (d) attack training set


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

through model updates sent to the server, the attacker introduces changes in the shared model $\theta_G^t$ that persist after a benign user fine-tunes $\theta_G^t$ 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 [5, 4]. Similarly to poisoning attacks in federated learning [6], after many meta-training steps on the local model $\theta_a^t$, the attacker sends a "boosted" update to the parameter server: $\delta_a^t = \lambda(\theta_a^t - \theta_G^t)$, where $\lambda$ is the *boosting factor* (to make it prevail over other updates).

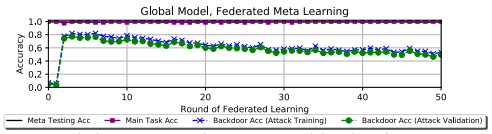
## 3 Effects of Backdoor Attacks

In this section, we explore backdoor attacks on the *Omniglot* [24] and *mini-ImageNet* [9, 25] datasets.

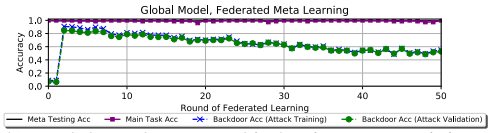
**Attack Evaluation.** We consider a federation of $M = 4$ users, where user $i = 1$ is the attacker and users $i = 2, 3, 4$ are benign; at each round, the server selects 3 users and waits for all of their updates (i.e., $M_{min} = M_r = 3$). The meta-model is initially trained only by benign users, reaching state-of-the-art accuracy; then, the attacker is selected *exactly once* (one-shot attack) and the poisoned update is boosted with $\lambda = 3$ [7, 6]. To evaluate the effectiveness of the attack, we generate $K$-shot, $N$-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.

**Omniglot.** This dataset consists of 1623 character classes from 50 alphabets, with 20 examples per class. Similarly to [11, 10, 26], we resize images to $28 \times 28$ and augment classes $4\times$ 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.

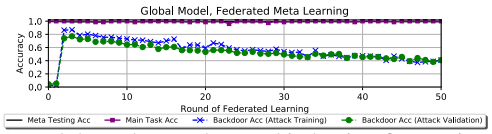
**mini-ImageNet.** This dataset includes 100 classes, each with 600 examples ($84 \times 84$ color images). We use 64 classes for meta-training (split among 4 users) and 20 classes for meta-testing, as in [25]; for each meta-training class, we hold out 20 validation examples. We also reserve 1 backdoor 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 validation sets, we use 100 and 50 additional examples, respectively, adding a backdoor key as in Fig. 2c-d.




(a) Backdoor examples not used by benign users

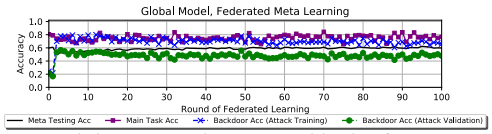



(b) Backdoor classes used in benign meta-training

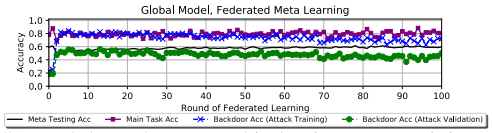



(c) Backdoor classes also used in benign fine-tuning

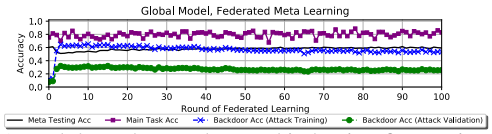
Figure 3: Benign meta-training after attacks on Omniglot




(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

Figure 4: Benign meta-training after attacks on mini-ImageNet

**Training Parameters.** All users run *Reptile* on the same Conv4 model as in [10, 11], a stack of $4$ modules ($3 \times 3$ Conv filters with batchnorm and ReLU) followed by a fully-connected and a softmax layer; the modules have $64$ filters and $2 \times 2$ max-pooling. We adopt the same parameters as in [11]: $5$-shot, $5$-way meta-testing of a meta-model trained with $E = 1000$ episodes $10$-shot, $5$-way (Omniglot) or $E = 100$ episodes $15$-shot, $5$-way (mini-ImageNet) per round at each user, with meta-batch size $B = 5$ and outer learning rate $\epsilon = 0.1$; for each episode, we use $e = 10$ (meta-training) or $e = 50$ (meta-testing) SGD steps, with inner batch size $b = 10$ and Adam optimizer ($\beta_1 = 0$, $\beta_2 = 0.999$, initial learning rate $\eta = 0.001$). The attacker trains for $E = 50000$ episodes and $50$ inner epochs (Omniglot), or $E = 150000$ and $1$ inner epoch (mini-ImageNet); backdoor and target examples $\mathcal{X}_B$ and $\mathcal{X}_T$ are always included by the attacker with 2:3 (Omniglot) or 1:2 (mini-ImageNet) ratio.

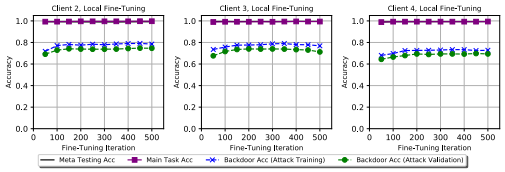
In our first set of experiments, benign users continue *federated meta-training* after the attack.

**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. 3a (Omniglot) and Fig. 4a (mini-ImageNet): before the attack (Round 0), meta-testing accuracy (black line) is above $99\%$ (Omniglot) or $60\%$ (mini-ImageNet); the attacker is selected in Round 1; then in Round 2 attack accuracy (classification of backdoor images as target class) reaches $78\%$ and $74\%$ on attacker's training dataset (blue line) and $77\%$ and $55\%$ on the held-out validation set (green line) for Omniglot and mini-ImageNet, respectively, while meta-testing accuracy on other classes remains above $98\%$ (Omniglot) or drops to $50\%$ (mini-ImageNet). Even after 50 (Omniglot) and 100 (mini-ImageNet) 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).

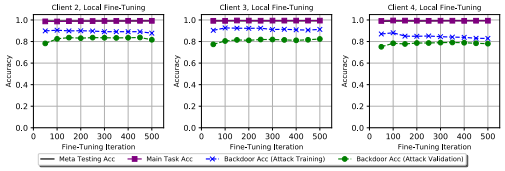
**Experiment 1(b).** Next, we consider the case where meta-training datasets of benign users include correctly-labeled images of backdoor classes during pre-training, so the meta-model should easily adapt to classifying them correctly. Results are in Fig. 3b (Omniglot) and Fig. 4b (mini-ImageNet): meta-testing accuracy is still above $98\%$ and $\approx 50\%$ after the attack for Omniglot and mini-ImageNet, respectively, while attack training/validation accuracy is close to $92\%$ / $83\%$ (Omniglot) and $76\%$ / $50\%$ (mini-ImageNet); after additional meta-training by benign users, attack training/validation accuracy is still $50\%$ / $50\%$ (50 rounds, Omniglot) and $69\%$ / $42\%$ (100 rounds, mini-ImageNet).

**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. 3c and 4c: after the attack (Round 2), meta-testing accuracy is still greater than $98\%$ (Omniglot) and $50\%$ (mini-ImageNet); however, attack training/validation accuracy drops to $90\%$ / $75\%$ (Omniglot) and $65\%$ / $32\%$ (mini-ImageNet), and, after additional meta-training by benign users, further drops to $40\%$ / $40\%$ (50 rounds, Omniglot), and $55\%$ / $25\%$ (100 rounds, mini-ImageNet), lower than previous results.

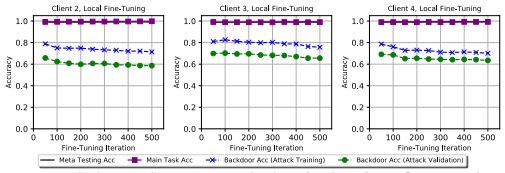
Overall, we observe that backdoor attacks are: (1) more successful on the attack training set (especially for mini-ImageNet), as expected; (2) similarly successful when benign users use correctly-labeled




(a) Backdoor examples not used by benign users

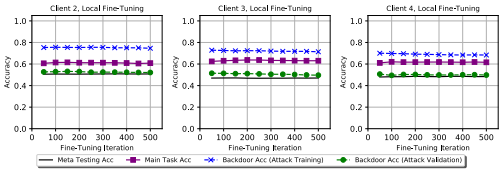



(b) Backdoor classes used in benign pre-training

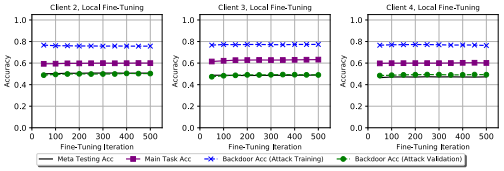



(c) Backdoor classes used also in benign fine-tuning

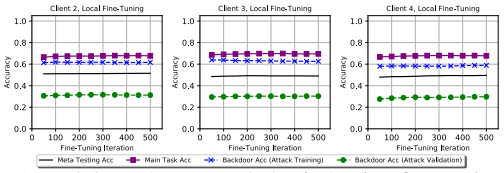
Figure 5: Benign fine-tuning ($\eta = 0.001$) after attacks on Omniglot




(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

Figure 6: Benign fine-tuning ($\eta = 0.001$) after attacks on mini-ImageNet

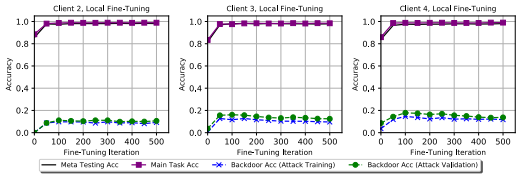
backdoor images for meta-training; (3) considerably less successful when fine-tuning also includes correctly-labeled backdoor images. Nonetheless, *it does not appear possible to rely only on additional meta-training to remove backdoor attacks.* In our next set of experiments, we explore whether *additional 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 $\eta = 0.001$, but run $e = 500$ ($10\times$ more) iterations of fine-tuning in Round 2 (right after the attack). Results are in Fig. 5 (Omniglot) and 6 (mini-ImageNet), 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. 5a and 5b); when backdoor classes are present (Fig. 5c), attack accuracy drops slightly for all users, and it is gradually reduced during fine-tuning ($\approx 10\%$ after $500$ iterations). For mini-ImageNet, when backdoor classes are not present during fine-tuning (Figs. 6a and 6b), 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. 6c), main-task accuracy is improved by $5\%$ and attack accuracy is reduced by $20\%$ for all users. From Fig. 5c and Fig. 6c, we observe that the presence of backdoor classes has limited influence on attack accuracy.

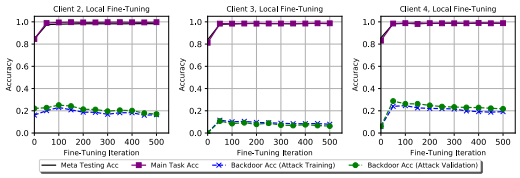
## 4 Matching Networks as a Defense Mechanism

Since defense mechanisms based on the analysis of updates received from users may violate privacy and are not compatible with secure update aggregation by the server, we propose a defense mechanism *applied locally by benign users*. The idea is inspired by *matching networks* [9], a popular meta-learning framework exploiting recent advances in attention mechanisms and external memories.

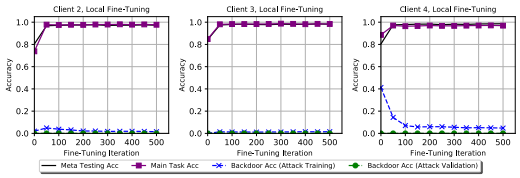
A matching network uses the output of an embedding model $f_\theta(x)$ to find similarities between input examples and reference examples from a *support set*. This non-parametric design, with external memories, allows matching networks to switch to a different classification task without supervised fine-tuning of $f_\theta$. Specifically, given the trained embedding model $f_\theta(x)$ and a support set $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^k$, class $\hat{y} = \arg\max_{i=1,\ldots,k} P(y_i|\hat{x}, \mathcal{S})$ is predicted where $P(y_i|\hat{x}, \mathcal{S})$ estimates output probabilities for the input $\hat{x}$. A common model is $\hat{y} = \sum_i a(\hat{x}, x_i) y_i$, a mixture of one-hot




(a) Backdoor examples not used by benign users

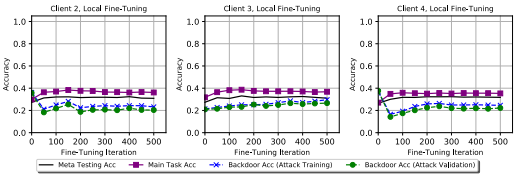



(b) Backdoor classes used in benign pre-training

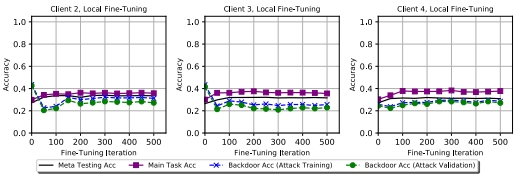



(c) Backdoor classes used also in benign fine-tuning

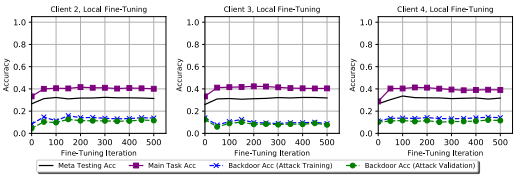
Figure 7: Benign fine-tuning of matching networks ($\eta = 0.001$) after attacks on Omniglot




(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

Figure 8: Benign fine-tuning of matching networks ($\eta = 0.001$) after attacks on mini-ImageNet

output vectors $y_i$ of the support set based on some *attention mechanism* $a(\hat{x}, x_i)$ [27, 28, 29, 30, 9, 31]. For example, $a(\hat{x}, x_i)$ can be a softmax over the cosine distance $c(\cdot, \cdot)$ of the embeddings of $\hat{x}$ and $x_i$, i.e., $a(\hat{x}, x_i) = e^{c(f_\theta(\hat{x}), f_\theta(x_i))} / \left( \sum_{j=1}^{k} e^{c(f_\theta(\hat{x}), f_\theta(x_j))} \right)$. We adopt a variant where (1) the output components of the embedding model $f_\theta$ are multiplied by gate variables $0 \leqslant \alpha_{l,j} \leqslant 1$, and (2) cosine distances to reference examples of each class are multiplied by a scaling factor $\beta_l$ [32]. 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 fine-tuning to train learnable parameters $\alpha_l$ and $\beta_l$. Before fine-tuning the adapted matching network, we apply a random Glorot initialization $\theta_g$ as $\theta' = \delta\theta + (1-\delta)\theta_g$ to reduce the influence of the poisoned model; then we train $\alpha_l$ and $\beta_l$ for a few iterations (with fixed $\theta'$), and finally train $\theta'$, $\alpha_l$ and $\beta_l$ jointly (training is performed as in [9, Sec. 4.1]). We use $\delta = 0.3$ and the same learning rate $\eta = 0.001$ of backdoor experiments in Section 3. 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.

**Experiment 3.** Results are reported in Figs. 7 and 8 (fine-tuning starts in Round 2, using the same parameters as in Experiment 2). *The proposed defense mechanism can successfully remove backdoor attacks:* when backdoor classes are not present in meta-testing (Figs. 7a, 7b, 8a and 8b), attack accuracy drops to $\approx 20\%$ (comparable to random assignment to one of the 5 classes) in a few epochs; when backdoor classes are present in meta-testing (Figs. 7c and 8c), attack accuracy significantly drops to $\approx 0\%$ (Omniglot) and $\approx 10\%$ (mini-ImageNet) in a few epochs of fine-tuning. Notably, meta-testing accuracy for Omniglot (Fig. 7) is always above $96\%$ after $50$ iterations; in contrast, meta-testing accuracy for mini-ImageNet (Fig. 8) is $\approx 35\%$, lower than in Fig. 6. This suggests a limitation of matching networks; other variants may overcome this limitation. An ablation study of our defense mechanism (not included due to space limitations) highlighted the importance of all of its elements (attention model, noisy meta-model initialization, fine-tuning procedure).

## 5 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.

## Acknowledgements

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

## Broader Impact

The context for this paper is Federated Learning (FL), a framework designed to allow users with insufficient but confidential data to jointly train machine learning models while preserving privacy. For example, a single hospital, clinic, or public health agency may not have sufficient data to train powerful machine learning models providing doctors with diagnostic aid (e.g., for medical images such as X-rays or fMRI images): FL allows multiple of these entities to jointly train more accurate machine learning models without sharing patients' private data. Unfortunately, FL is known to be vulnerable to poisoning backdoor attacks, where a malicious user of the federation can control the decisions of the model trained jointly with benign users. In the context of diagnostic medical aid, backdoor attacks to FL could mislead doctors into making incorrect diagnoses, with life threatening risks; similarly, consequences of backdoor attacks could be disastrous for many applications in healthcare, transportation, finance. Although defense mechanisms have been proposed in the literature, state-of-the-art solutions against poisoning backdoor attacks rely on *third parties to examine user updates to the model*, violating the fundamental *privacy* motivation of FL. We believe that an effective *user-end defense mechanism* can guard against backdoor attacks while preventing unexpected abuse due to privacy leaks. Thus the broader impact of our proposed approach is that it can prevent attacks on machine learning models that are developed jointly by multiple entities as well as privacy-related abuse. Allowing multiple entities to jointly develop machine learning models is critical to the broader impact of machine learning applications in settings (e.g., healthcare) where data is scarce and sensitive.

## References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282, 2017.

[2] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy preserving machine learning," *IACR Cryptology ePrint Archive*, vol. 2017, p. 281, 2017.

[3] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Trans. Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2018.

[4] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *CoRR*, vol. abs/1712.05526, 2017.

[5] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *CoRR*, vol. abs/1708.06733, 2017.

[6] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. B. Calo, "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*, pp. 634–643, 2019.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," *CoRR*, vol. abs/1807.00459, 2018.

[8] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems 32, NIPS 2019, December 8-14, 2019, Vancouver, Canada*, pp. 8632–8642, Curran Associates, Inc., 2019.

[9] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems 29, NIPS 2016, December 5-10, 2016, Barcelona, Spain*, pp. 3630–3638, 2016.

[10] C. Finn, P. Abbeel, and S. Levine, "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 August 2017*, pp. 1126–1135, 2017.

[11] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *CoRR*, vol. abs/1803.02999, 2018.

[12] F. Chen, Z. Dong, Z. Li, and X. He, "Federated meta-learning for recommendation," *CoRR*, vol. abs/1802.07876, 2018.

[13] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," *CoRR*, vol. abs/1909.05830, 2019.

[14] S. Shen, S. Tople, and P. Saxena, "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, December 5-9, 2016*, pp. 508–519, 2016.

[15] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems 30, NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 119–129, 2017.

[16] J. Steinhardt, P. W. Koh, and P. Liang, "Certified defenses for data poisoning attacks," in *Advances in Neural Information Processing Systems 30, NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 3517–3529, 2017.

[17] M. Qiao and G. Valiant, "Learning discrete distributions from untrusted batches," in *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pp. 47:1–47:20, 2018.

[18] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "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*, pp. 5636–5645, 2018.

[19] C. Xie, O. Koyejo, and I. Gupta, "Generalized byzantine-tolerant SGD," *CoRR*, vol. abs/1802.10116, 2018.

[20] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems 31, NIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 8011–8021, 2018.

[21] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," in *Proceedings of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2018, Irvine, CA, USA, June 18-22, 2018*, p. 96, 2018.

[22] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "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*, pp. 3518–3527, 2018.

[23] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *CoRR*, vol. abs/1808.04866, 2018.

[24] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[25] Q. Sun, Y. Liu, T. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. P. Lillicrap, "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*, pp. 1842–1850, 2016.

[27] D. Bahdanau, K. Cho, and Y. Bengio, "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*, 2015.

[28] T. Luong, H. Pham, and C. D. Manning, "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, September 17-21, 2015*, pp. 1412–1421, 2015.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30, NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

[30] M. Collier and J. Beel, "Implementing neural turing machines," in *27th International Conference on Artificial Neural Networks, ICANN 2018, Rhodes, Greece, October 4-7, 2018, Proceedings, Part III*, pp. 94–104, 2018.

[31] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems 30, NIPS 2017, 4-9 December 2017, Long Beach, CA, USA*, pp. 4077–4087, 2017.

[32] W. Chen, Y. Liu, Z. Kira, Y. F. Wang, and J. Huang, "A closer look at few-shot classification," *CoRR*, vol. abs/1904.04232, 2019.