Dynamic backdoor attacks against federated learning

Anbu Huang
WeBank AI Lab
Shenzhen, China
stevenhuang@webank.com

ABSTRACT
Federated Learning (FL) is a new machine learning framework, which enables millions of participants to collaboratively train machine learning model without compromising data privacy and security. Due to the independence and confidentiality of each client, FL does not guarantee that all clients are honest by design, which makes it vulnerable to adversarial attack naturally. In this paper, we focus on dynamic backdoor attacks under FL setting, where the goal of the adversary is to reduce the performance of the model on targeted tasks while maintaining a good performance on the main task, current existing studies are mainly focused on static backdoor attacks, that is the poison pattern injected is unchanged, however, FL is an online learning framework, and adversarial targets can be changed dynamically by attacker, traditional algorithms require learning a new targeted task from scratch, which could be computationally expensive and require a large number of adversarial training examples, to avoid this, we bridge meta-learning and backdoor attacks under FL setting, in which case we can learn a versatile model from previous experiences, and fast adapting to new adversarial tasks with a few of examples. We evaluate our algorithm on different datasets, and demonstrate that our algorithm can achieve good results with respect to dynamic backdoor attacks. To the best of our knowledge, this is the first paper that focus on dynamic backdoor attacks research under FL setting.

CCS CONCEPTS
• Computing methodologies → Artificial intelligence; Machine learning; Distributed computing methodologies.

KEYWORDS
Federated Learning, Meta Learning, Adversarial Machine Learning, Privacy Preserving Machine Learning

1 INTRODUCTION
In the past decade, deep learning had shown powerful representation and learning capabilities [13], coupled with increasing amount of data and computational power, which made deep learning achieve unprecedented success in many commercial applications, such as computer vision [9, 12, 20], nature language processing [5, 17, 25], speech recognition [8, 24], etc. Traditional machine learning process requires centralizing of the training data on one machine, however, this learning paradigm had been proven unsafe and vulnerable to data leakage [16]. Besides that, following privacy concerns among users and governments, policy makers have responded with the implementation of data privacy legislations such as General Data Protection Regulation (GDPR) [27] and California Consumer Privacy Act (CCPA), which prohibit data upload without user’s permission explicitly.

To decouple the need for model training with the need to store the data in the cloud or central database, a new distributed learning paradigm, called federated learning, was introduced in 2016 by Google [16]. In contrast to the centralized machine learning approaches, FL distributes the machine learning process over to the edge, and enables each client to collaboratively learn a shared model while keeping the training data on device, this strategy can significantly mitigate many of the systemic privacy risks, and has been widely used in high privacy requirements areas, such as financial [30], healthcare [14], computer vision [15], etc.

In spite of this, since FL does not guarantee that all clients are honest by design, which makes it vulnerable to adversarial attack, in this paper, we focus on backdoor attacks, one of the most popular attacks in adversarial machine learning, where the goal of the attacker is to reduce the performance of the model on targeted tasks while maintaining a good performance on the main task, e.g., the attacker can modify an image classifier so that it assigns an attacker-chosen label to images with certain features [1, 29].

Current existing studies are mainly focus on static and specific adversarial targeted attacks, however, as we all know, FL is an online learning framework, the attacker can choose new attack target on the fly, to avoid learning from scratch, we propose a novel FL algorithm, which can train a versatile model to fit both targeted task and main task on one hand, and fast adapt to new targeted tasks on the other hand, our approach connect meta-learning with backdoor attacks, the algorithm workflow as shown in figure 1, here, we regard online FL training as a series of episodes, each episode represents one FL training stage, Jiang et al. [?] pointed out that optimization-based meta-learning algorithm can be seen as a special implementation of FL, which makes meta-learning well suited for implementation on FL framework.

we summarize our main contributions as follows:

• We shed light on an important problem that has not been studied so far, to the best of our knowledge, this is the first paper that focus on dynamic backdoor attacks under FL setting.
• We propose a new framework, called symbiosis network, for malicious client’s local model training, we point out that this strategy can make backdoor attack more persistent with respect to adversarial backdoor attack.
• We connect meta-learning with backdoor attacks under FL setting, and give an simple implementation, our algorithm only need to make slightly modifications to existing federated averaging algorithm.
• We provide a comprehensive theoretical analysis of dynamic backdoor attacks under FL setting, and raise three objectives which are need to be solved for this type of problem.
Figure 1: Schematic illustration of dynamic backdoor attack. Suppose we have two malicious clients: \( C_1 \) and \( C_2 \), all malicious client’s datasets contain two parts: clean dataset (\( D_{\text{cln}} \)) and adversarial (poisoned) dataset (\( D_{\text{adv}} \)). We treat each local model update as an individual task (\( T_i \)). Here, \( T_1 \) and \( T_2 \) are backdoor attack tasks. After episode 1 done, \( C_1 \) changes poisoned datasets and makes the backdoor task change from \( T_1 \) to \( T_1' \), to avoid learning from scratch, our initial model should utilize previous experiences (episode 1), and quickly adapt to new poisoned datasets.

2 BACKGROUND AND RELATED WORKS

In this section, we briefly review the background of related works, including federated learning, federated meta-learning and backdoor attacks against federated learning.

2.1 Federated Learning

Traditional machine learning approach requires raw datasets uploaded and processed centrally, however, due to data privacy and security, sending raw data to the central database is regarded as unsafe, and violate the General Data Protection Regulation (GDPR). To decouple the need for model training with the need to store the data in the central database, a new machine learning framework called federated learning was proposed, a typical FL framework is as shown in figure 2.

In FL scenario, each client update their local model based on local datasets, and then send the updated model’s parameters to the server side for secure aggregation, these steps are repeated in multiple rounds until the learning process converges.

Suppose \( C = \{C_1, C_2, ..., C_N\} \) represent all client sets, \( S \) refers to server, when each round begins, the server selects a subset of devices, and send initial model to these clients, generally speaking, standard FL procedure including the following three steps [3]:

- **Local Model Training**: Denote \( t \) as the current iteration round, \( C_i \) represents client \( i \) (\( i = 1, 2, ..., N \)), \( N \) is the number of clients, \( G_i^t \) (\( G_i^t = G_i^t \)) and \( D_i \) represent the local model and local dataset of client \( i \) respectively. Based on \( D_i \), each client update the local model from \( G_i^t \) to \( L_i^{t+1} \) respectively, then send the updated local model parameters \( L_i^{t+1} - G_i^t \) to the server side for aggregation.

- **Global Aggregation**: The server side collect the updated parameters from selected clients, and do model aggregation to obtain the new joint model:

\[
G_i^{t+1} = G_i^t + \eta \sum_{i=1}^{m} (L_i^{t+1} - G_i^t)
\]

where \( \eta \) represents the factor which controls the fraction of the joint model, specifically, if \( \eta = 1 \), equation 1 is equal to weight average.

- **Update Local Model**: When the aggregation is completed, the server side select a subset of clients again, and send global model \( G_i^{t+1} \) back to the selected clients for next iteration and repeat this cycle until converge.

Figure 2: Federated Learning Architecture
2.2 Federated Meta-Learning

Meta-learning, also known as “learning to learn”, is aimed to learn a versatile model from a variety of tasks, so that it can be quickly adapted to new tasks with few training examples. Meta-learning have typically fallen into one of three categories: metric-based [11, 21, 23, 26], model-based [7, 28], and optimization-based [6, 18], in this paper, we only consider optimization-based meta-learning algorithm.

Optimization-based meta-learning algorithm seeks an initialization for the parameters of a neural network, such that the network can be fine-tuned using a small amount of data from a new task and few gradient steps to achieve high performance. Typical optimization-based meta-learning algorithm can be decomposed into the following two stages [31]:

- **Inner Update**: for a given task $T_i$, with corresponding loss $L_{T_i}$, the inner-loop performs stochastic gradient descent to optimize loss function to get optimal parameters for task $T_i$.

$$\theta^*_i = \arg \min_{\theta} L_{T_i}(D_i^{\text{train}}, \theta)$$

- **Outer Update**: the outer loop perform meta optimization. We first sample batch of tasks $T_i$, where $T_i \sim p(T)$, the objective of meta learner is to achieve a good generalization across a variety of tasks, we would like to find the optimal parameters, such that the task-specific fine-tuning is more efficient, this leads us to the following objective function for outer update:

$$\theta = \min E_{T_i \sim p(T)} \left\{ L_{T_i}(D_i^{\text{test}}, \theta^*_i) \right\}$$

Jiang et al. [20] pointed out that optimization-based meta-learning can be seen as a special implementation of FL, and FL as a natural source of practical applications for MAML algorithms [6]. Chen et al. [4] propose a federated meta-learning framework, called FedMeta, to improve personalized recommendation, where a parameterized algorithm (or meta-learner) is shared, instead of a global model in previous approaches.

2.3 Backdoor attacks against federated learning

Backdoor attack is one of the most popular attacks of adversarial machine learning, the attacker can modify or fool an image classifier so that it assigns an attacker-chosen label to images with certain features. Some examples are as shown in figure 4.

As previously mentioned in abstract, FL does not guarantee that all clients are honest by design, and hence makes it vulnerable to adversarial attack naturally. Backdoor attack under FL setting had been studied extensive [1, 2, 22, 29], however, unlike distributed machine learning, backdoor attack under FL setting is much harder than what we thought, the main reason is that FL requires the server selects a subset of (not all) connected devices at each round for model training, if attackers only control a small number of malicious agents, the probability of being selected of each round could be low, which leading aggregation cancels out most of the malicious model’s contribution and the joint model quickly forgets the backdoor.

To make backdoor attack more effective and persistent, one feasible solution is using explicit boosting strategy, that is to say, adversaries scale up the weights of the poisoned model to ensure that the backdoor attack survives the averaging. Xie et al. [29] proposed distributed backdoor attack, which decomposes a global trigger pattern into separate local patterns, and distributed these local trigger patterns to different malicious clients, this strategy shows more persistent and stealthy than centralized backdoor attack.

Current approaches are mainly focus on static attack, in this paper, what we concern about is dynamic backdoor attack, a concrete example is shown in figure 3. At episode 1, attacker $C_1$ embeds text data (“KDD”) in the image as poisoned dataset (labeled as “dog”) but ground-truth is “fish”), after collaboratively train a new global model, it can identify images containing "KDD" text as "dog”, and not affect normal image classification; at episode 2, $C_1$ changes embedded text data (“ACM”) in the image as new poisoned dataset (labeled as “spider” but ground-truth is "dog"), new aggregated model should identify this new pattern correctly.

3 DYNAMIC BACKDOOR ATTACKS VIA META-LEARNING

In this section, we will define the problem definition, present the general ideas and theoretical analysis of our algorithm.

3.1 Attacker ability setting

In this paper, we suppose attackers fully control a subset of clients, malicious clients are non-colluding with each other. according to literature [10], we can summarize attacker ability in table 1.

3.2 Dynamic backdoor attacks problem set up

Federated learning, as an online learning framework, the targeted task can be changed dynamically by attacker, compared with static backdoor attacks, dynamic scenario poses more difficulties and challenges during model training, which leads us to first introduce the following three objectives for dynamic backdoor attacks, for the sake of consistence in this paper, we will reuse symbol definitions of section 2.1 in the following discussion.

Obj 1: Achieve high performance on both main task and backdoor task.

Let’s define $C_i$ represents client $i$, each client keep dataset $D_i$ on device locally, for malicious client $C_i$, $D_i$ consists of two parts: clean dataset $D_i^{\text{clean}}$ and adversarial (poisoned) dataset $D_i^{\text{adv}}$. $D_i^{\text{clean}}$ and $D_i^{\text{adv}}$ should satisfied:

$$D_i^{\text{clean}} \cap D_i^{\text{adv}} = \phi, \quad D_i^{\text{clean}} \cup D_i^{\text{adv}} = D_i$$

(4)

To achieve high performances on both tasks, our goal is to train appropriate model parameters so that it can make good predictions in both clean and poisoned datasets, this implies the following objective equation for client $C_i$ in round $t$ with local dataset $D_i$:

$$\theta_i^* = \arg \max_{\theta} \left\{ \sum_{j \in D_i^{\text{clean}}} \left[ p(L_i^{t+1}(x_j; \theta) = y_j) \right] + \sum_{j \in D_i^{\text{adv}}} \left[ p(L_i^{t+1}(x_j; \theta) = \pi_j) \right] \right\}$$

(5)
Figure 3: An concrete example of dynamic backdoor attacks, currently, we have four clients, only $C_1$ is malicious, attacker create adversarial examples by injecting poison (embed text "KDD" into the image) in episode 1, after that, the attacker injects new poison pattern (embed text "ACM" into the image) for model training in episode 2.

Table 1: summary of attack ability setting in our paper

| Characteristic          | Setting        | Description                                                                                                                                 |
|-------------------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Attack vector           | Poisoning attack| The attacker can fully control malicious client, that is to say, (a). the adversary can alter the client datasets used to train the model; (b). the adversary can alter model update strategies, such as model parameters and loss function. |
| Knowledge               | White box      | The adversary has the ability to directly inspect the parameters of the model.                                                                 |
| Participant collusion   | Non-colluding  | There is no capability for participants to coordinate an attack.                                                                                |
| Participation rate      | Dynamic        | A malicious client participates in local model training if and only if it was selected by the server.                                         |
| Adaptability            | Dynamic        | Adversarial targets can be changed dynamically by attacker.                                                                                    |

Here, we decompose the right side of equation 5 into two parts,
- the first part represents training on clean dataset $D^i_{cln} = (x^i, y^i)$, optimizing this part can make good performance on main task.
- the second part represents training on poisoned dataset $D^i_{adv} = (x^i, \pi^i)$, where $\pi^i$ is attacker-chosen label, optimizing this part can make good performance on targeted backdoor task.

Obj 2: Persistent, robustness and stealthy

As we have discussed in section 1, under FL setting, a subset of clients are chosen at each round randomly, which means that we do not guarantee malicious clients could be selected every time, if that is the case, model aggregation at server side can cancel out most of the malicious model’s contribution and the joint model quickly forgets the backdoor.
To make our algorithm more persistent, robustness and stealthy, we propose symbiosis network, a new local model training paradigm for FL.

**Symbiosis Network**: In the standard FL scenario, when every round local training begins, we need to first replace local model with global model, this could be make sense since global model contains rich hidden features which are derived from data scattered across clients, however, under dynamic backdoor attacks setting, attacker may inject new training samples which are completely different from the previous data distribution, replacing local model with global model may degrade model performance.

![Figure 5: The architecture of symbiosis networks for local model training](image)

For these reasons, we propose a new training architecture for malicious clients, called **symbiosis network**, as shown in figure 5. We take classification as an example, for malicious client \(C_i\), we keep local model and global model simultaneously, and modify local model training objective function as follows:

\[
\mathcal{L} = (1 - p) \cdot \mathcal{L}_{\text{class}} + p \cdot \mathcal{L}_{\text{dist}}
\]  

(6)

Figure 5: The architecture of symbiosis networks for local model training

Here, \(\mathcal{L}_{\text{class}}\) captures the accuracy on both the main and backdoor tasks. \(\mathcal{L}_{\text{dist}}\) calculate the distance between local model and global model. this objective function is similar to the approach proposed by Bagdasaryan et al. [1] and Xie et al. [29], however, the essential different is that, [1, 29] set the factor \(p\) manually, and find the optimal value through trial and error strategies, while in our approach, we notice that \(\mathcal{L}_{\text{class}}\) and \(\mathcal{L}_{\text{dist}}\) have different contribution throughout model training, \(p\) is the factor to balance this contribution, one feasible choice is to set \(p\) as model performance of global model, for classification tasks, \(p\) is equal to classification accuracy. We can verify the rationality of our approach by the following three aspects:

1) if \(p\) is large, it means that global model can achieve good results on new adversarial examples, our goal is to make the local model as close to the global model as possible, therefore, minimizing \(\mathcal{L}_{\text{dist}}\) is the main contribution of loss function \(\mathcal{L}\). Specifically, if \(p = 1.0\) (perfect prediction for new poisoned datasets), minimize \(\mathcal{L}\) is equal to minimize \(\mathcal{L}_{\text{dist}}\).

2) if \(p\) is small, it means that global model has poor performance on new adversarial examples, global model could be far away from optimal parameters, therefore, minimizing \(\mathcal{L}_{\text{class}}\) is the main contribution of loss function. Specifically, if \(p = 0.0\) (terrible prediction for new poisoned datasets), minimize \(\mathcal{L}\) is equal to minimize \(\mathcal{L}_{\text{class}}\).

3) [1, 29] set the factor \(p\) manually, which means that \(p\) is fixed throughout the training process, it is not flexible, and is easy to diverge or stuck at local optimal point.

**Obj 3: Fast adaptation to new targeted task**

The objective of dynamic backdoor attacks is not just to make good performances for specific targeted task, but also to fully exploit previous experiences and quickly adapt to new task, for this purpose, the global model need to learn an internal feature that is broadly applicable to all tasks, rather than a single task. we can achieve this objective by minimizing the total loss across tasks sampled from the task distribution:

\[
\mathcal{L} = \min_{\hat{\theta}} \sum_{T_i \sim p(T)} L_{T_i}(f_{\theta^*})
\]

(7)

Here, \(\theta^*\) is the optimal parameters for task \(T_i\), solved by equation 5, figure 6 gives us a more intuitive illustration, figure 6(a) shows that normal FL need to learn new task from scratch, and take many SGD steps to converge; on the other hand, figure 6(b) makes use of previous experiences, so that the initial model parameters \(\theta^*\) much more closer to each task’s optimal parameters than \(\theta\), only a few SGD steps can guarantee convergence.

![Figure 6: Comparison of normal task training and task training via meta-learning](image)

The optimization problem of equation 7 is the same as MAML [6], however, optimize equation 7 will cause two problems, and hence make it hard to apply to federated learning.

- optimize equation 7 will cause second-order derivatives, and make the computation expensive.
- the optimization requires keeping additional datasets to update at server side, which violate data privacy and security.

To this end, We solve this problem with another way, since our goal is to learn an internal feature that is broadly applicable to all tasks, this equal to the fact that global model parameter should close to each task’s optimal parameters with some distance metrics, if we use euclidean distance as our distance measure, this motivate us to deduce the following new loss function:
We create a federated learning cluster consists of one server and 100 clients, among of them, 6 of whom are malicious clients, the dynamic injected poison pattern for each malicious client is shown in figure 7.

Without loss of generality, we set $C_i$ ($i = 1, 2, ..., 6$) are malicious clients, the initial poison patterns for each malicious client are listed in table 2. We split datasets using dirichlet distribution and assign them to each client respectively, for malicious clients, about 12 percent are poison datasets.

We choose three popular image datasets to evaluate our approach, including mnist, cifar-10 and tiny imagenet. These three datasets are used to train a classification model. The experimental results show that our approach is effective in defending against backdoor attacks.
datasets are increasing in size and are therefore good candidates for comparison.

4.2 Evaluation on performance and persistent
As shown in figure 8, we run three different CNN architecture (LeNet for MNIST, ResNet for cifar-10 and DenseNet for tiny imagenet) to evaluate performance and persistent (see section 3.2).

Figure 8(a), 8(b), 8(c) shown the backdoor accuracy performance, As previous mentioned, backdoor attack under FL setting is much harder than what we thought, the main reason is that model aggregation would cancel out most of the malicious model’s contribution and the joint model quickly forgets the backdoor, the fluctuations in the graph are due to the impact of model aggregation, we compare manually setting $p$ value [1, 29] with symbiosis network training (see equation 6), our symbiosis network training outperform manually setting approach in most case with respect to backdoor accuracy, besides that, as the iteration progresses, this advantage can be maintained, which means that our attack approach is persistent and robust.

Figure 8(d), 8(e), 8(f) shown the main task accuracy performance of our approach, as we can see, backdoor attack does not significantly affect the main task, and achieve good performances in all three datasets.

4.3 Evaluation on fast adaptation
We use meta optimization describe in equation 8 as our aggregation to improve model adaptation capability, and make it quickly adapt to new poisoned task. To simulate this process, we use initial injected poisons (see table 2) for malicious clients in episode 1, after that, we inject new embedded text “KDD” into local images of client $C_1$, and use it as our new poisoned datasets in episode 2.

Here, we use federated averaging algorithm as our baseline, the performance is shown in figure 9, after the first few rounds, the meta-learning method quickly surpassed the federated averaging and achieve the same results with fewer steps.

5 CONCLUSION AND FUTURE WORKS
Federated learning is appealing because of its confidentiality and scalability, although adversarial attacks under federated learning setting has been studied extensively, it is still mainly focus on static scenarios. Dynamic backdoor attacks, on the other hand, are more challenging and ubiquitous in our real world.

In this paper, we introduce dynamic backdoor attacks problem under federated learning setting, and propose three corresponding objectives, coupled with detailed definitions and solutions for each of them, finally, we give an efficient and feasible solution to solve this problem. In future work, We intend to improve our work from the following two aspects:

- Our experiments mainly focus on image classification problems, we will verify the correctness of our algorithm with more experimental results.
- Explore how to improve other aggregation algorithms so that it can be compatible with meta-learning framework.
REFERENCES

[1] Eugene Bagdasaryan, Andreas Vei, Yaqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2018. How To Backdoor Federated Learning. CoRR abs/1807.00459 (2018). arXiv:1807.00459
[2] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Serapkin B. Caco. 2018. Analyzing Federated Learning through an Adversarial Lens. CoRR abs/1811.12470 (2018). arXiv:1811.12470
[3] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dmitry Huha, Alex Ingerman, Vladimir Ivanov, Chlof Kiddon, Jakub Koncynec, Stefano Maizocchi, H. Brendan McMahan, Timon Van Overveldt, David Petrov, Daniel Ramage, and Jason Roseland. 2019. Towards Federated Learning at Scale: System Design. CoRR abs/1902.01046 (2019). arXiv:1902.01046
[4] Fei Chen, Zhennhua Dong, Zhenguo Li, and Xuqiang He. 2018. Federated Meta-Learning for Recommendation. CoRR abs/1802.07876 (2018). arXiv:1802.07876 http://arxiv.org/abs/1802.07876
[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. CoRR abs/1810.04805 (2018). arXiv:1810.04805
[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. CoRR abs/1703.03400 (2017). arXiv:1703.03400 http://arxiv.org/abs/1703.03400
[7] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing Machines. http://arxiv.org/abs/1410.5401. cite arxiv:1410.5401.
[8] Anni Y. Hannek, Carl Case, Jared Casper, Bryan Catanzano, Greg Diamos, Ezhil Elsen, Ryan Prenge, Sanjeev Satheshe, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. CoRR abs/1412.5567 (2014). arXiv:1412.5567
[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. CoRR abs/1512.03385 (2015). arXiv:1512.03385
[10] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurelien Bellet, Melodi Bengio, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019).
[11] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese Neural Networks for One-shot Image Recognition.
[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey H. Hinton. 2012. ImageNet Classiﬁcation with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (Lake Tahoe, Nevada). Curran Associates Inc., 1097–1105.
[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep Learning. Nature 521, 7553 (2015), 436–444. https://doi.org/10.1038/nature14539
[14] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sebastien Ourselin, M Jorge Cardoso, et al. 2019. Privacy-preserving Federated Brain Tumour Segmentation. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 521–531. https://doi.org/10.1038/nature14539
[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. cite arxiv:1301.3781.
[16] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On First-Order Meta-Learning Algorithms. CoRR abs/1803.02999 (2018). arXiv:1803.02999 http://arxiv.org/abs/1803.02999
[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8026–8037. http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[18] Joseph Redmon and Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. arXiv preprint arXiv:1804.02767 (2018).
[19] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. CoRR abs/1703.05175 (2017). arXiv:1703.05175 http://arxiv.org/abs/1703.05175
[20] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. 2019. Can You Really Backdoor Federated Learning? arXiv preprint arXiv:1901.07963 (2019).
[21] Paul Voigt and Axel von dem Bussche. 2017. The EU General Data Protection Regulation (GDPR): A Practical Guide (1st ed.). Springer Publishing Company, Incorporated.
[22] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory Networks. (2014). http://arxiv.org/abs/1410.3916 cite arxiv:1410.3916.
[23] Chulun Xie, Keli Huang, Pin-Yu Chen, and Bo Li. 2020. DBA: Distributed Backdoor Attacks against Federated Learning. In International Conference on Learning Representations. https://openreview.net/forum?id=rkgx50YFr
[24] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated Machine Learning: Concept and Applications. CoRR abs/1902.04885 (2019). arXiv:1902.04885
[25] Luisa Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. 2019. Fast Context Adaptation via Meta-Learning. In Thirty-sixth International Conference on Machine Learning (ICML).