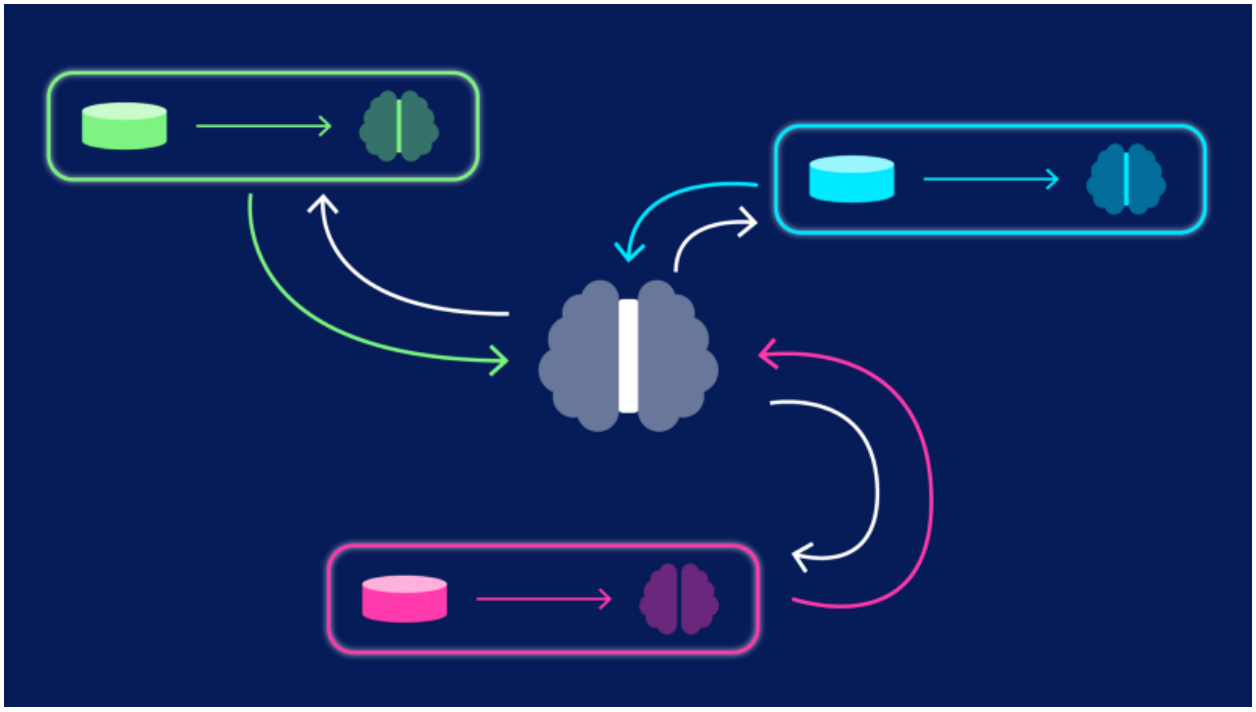
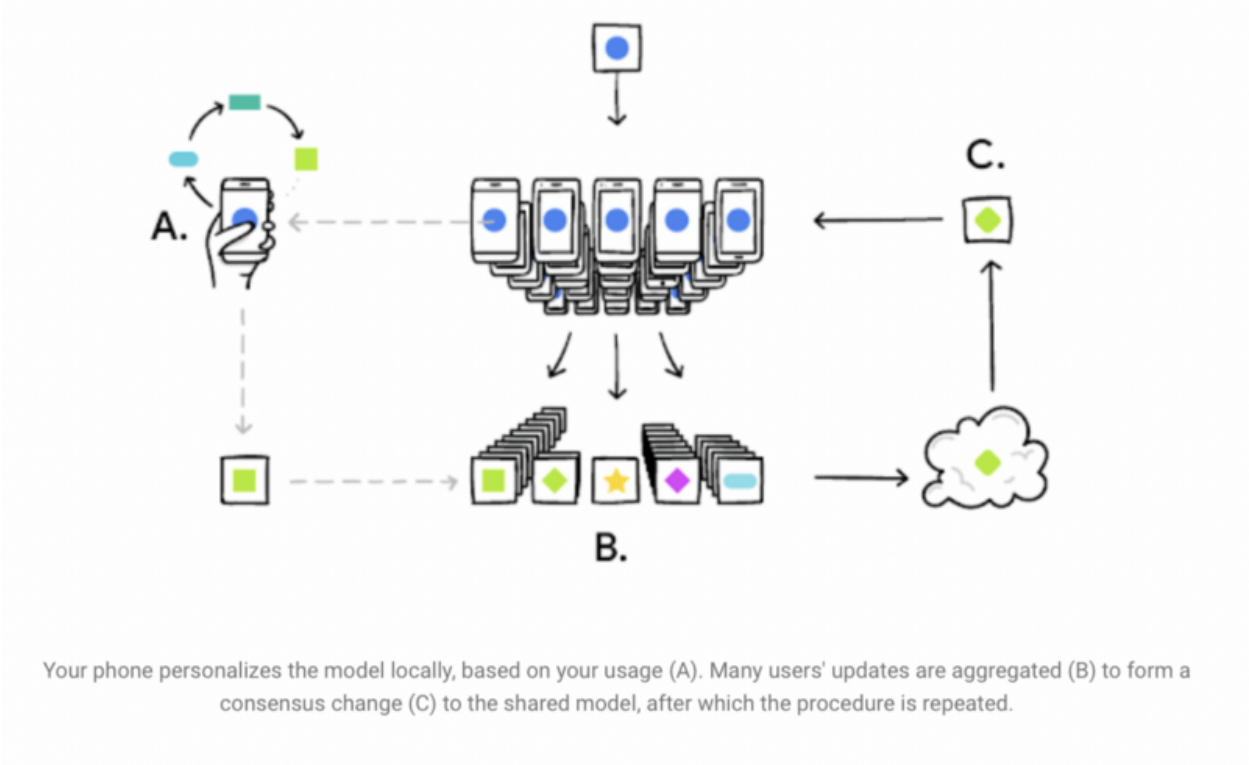
Federated Learning—Are We Safe?



Federated learning high overview diagram[12]

Federated Learning has become a very important paradigm in Machine Learning as data privacy has become an important factor for users. But despite its goal, how secure is Federated Learning from vulnerabilities and attacks from malicious users? And what is the future of Federated Learning? Before we dive in, let's first get a quick refresh on what Federated Learning is.

Why, What, and How Federated Learning?



Your phone personalizes the model locally, based on your usage (A). Many users' updates are aggregated (B) to form a consensus change (C) to the shared model, after which the procedure is repeated.

[2]

Federated learning has been surfacing up to tackle the issue with privacy. In this system, a model is stored on a centralized server. This is the global model which will update all the devices (think mobile phones, laptops etc) in the system with the latest and highly accurate model.[1]

Because this is Federated Learning, we cannot have training data on the centralized server. Malicious users could penetrate into the server and steal this data, rendering the main idea of “federated” utterly ineffective.

So what data does this model train on and how? The global model is first downloaded by a user on their device. The model is then trained on the relevant data available on the device. Once the training process completes, it will update back the global model with what it has learned. The global model will then aggregates the results from all the devices to enhance the accuracy of the model, which is then redeployed to the devices.[2] This iterative process continues over time, benefitting the model and users with highly accurate results.

What’s The Issue Here?

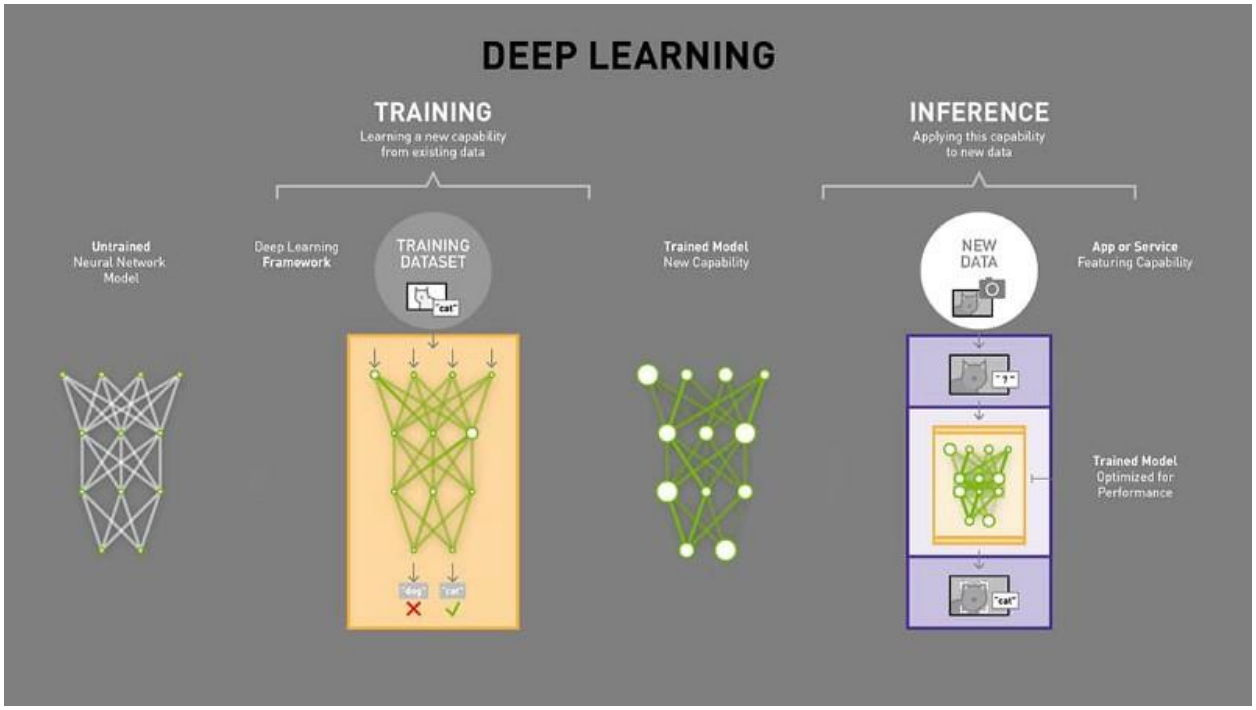
You might be thinking “Seem’s pretty secure to me. Federated Learning is awesome”. My data, which the model uses for training, stays on my device and the model is hosted on a centralized server. Even if an intruder were to attack the centralized server, the only thing they could gain is the model. What gives?

The model is the vulnerable piece here. Remember this is the same model which was created and trained on your data. It’s completely possible that the information from your data still lies within the model. Though it might be difficult to decrypt, that won’t stop an attacker from trying.[1]

[3]

What Kind Of Threats Are We Talking?

Threats to the model occur either at the **training** or **inferencing** phase.



Visualize the phases and how they are connected[6]

The **training phase** is when the model tries to learn rules and patterns from the data which it will ultimately use to make accurate predictions. During the training phase, malicious attackers can either benefit from the model by learning, influencing, or damaging the model.[1] Typically malicious attackers engage in **Poison Attacks**. This consist of attacks on the integrity of the training data collection or the learning process of the model.[1] Inference attacks can be be made during the training phase (inference ***attack*** vs inference ***phase***, two different things) on individual updates or on the aggregation of all updates from the users. Essentially, malicious attackers with high confidence can gain information about the users.[4]

Though we just spoke of inference ***attacks*** in the training phase, this is different from the inference ***phase***. The **inference phase** is when the model is able to use what it learned from the training phase to make predictions, in other words “infer”.[5] Malicious attackers don’t interfere with the model here. Attackers here can cause the model to give incorrect predictions. Or learn information about the model which can help them exploit the model. The effectiveness of this attack really depends on the the type of access the attacker has (full access or just being able to query the model).[1] Because of the design of Federated Learning, attackers can access user’s devices when the global model is attempting to deploy the updated model to all the clients.[1]

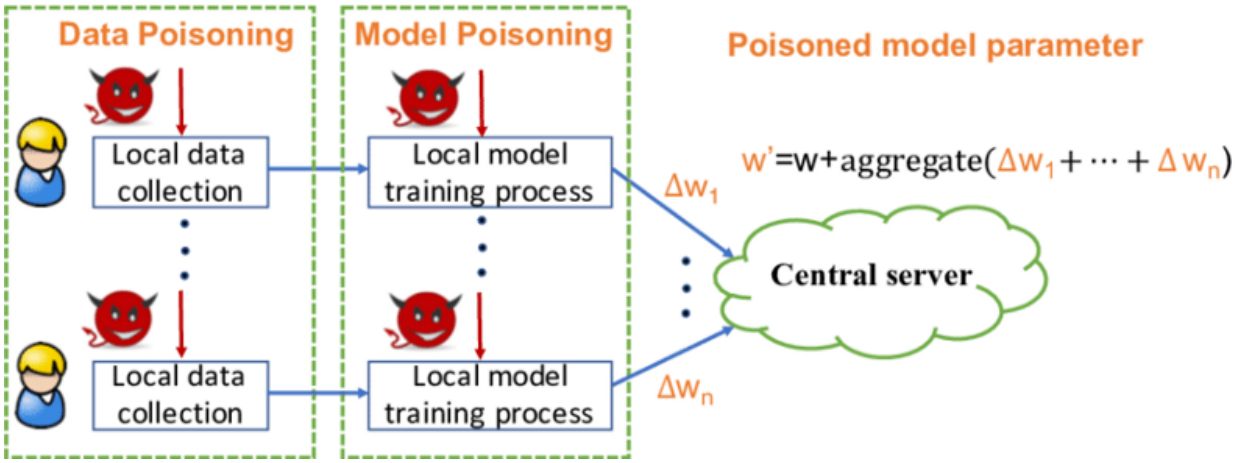
Now that we have a rough overview, let’s take a look into the types of attacks in each phase. **In this article we will focus only on training phase attacks.**

Training Phase Attacks

1. **What Are Poison Attacks?**

These attacks come in two forms, random or targeted. In a random attack, the main goal is to reduce the accuracy of the model.[1] Targeted attacks are executed to influence the outcome of the model to benefit the malicious attacker. Though targeted attacks are more appealing and beneficial to a malicious attacker, these prove to be more difficult as the malicious attack is trying to achieve a more specific goal.[1]

These poison attacks are performed either on the **data** (local data collection) or the **model** (local model training).[1] Regardless of the attack, the main goal is to change the underlying behavior of the model in an intolerable process.



Visualizing data and model poisoning process[1]

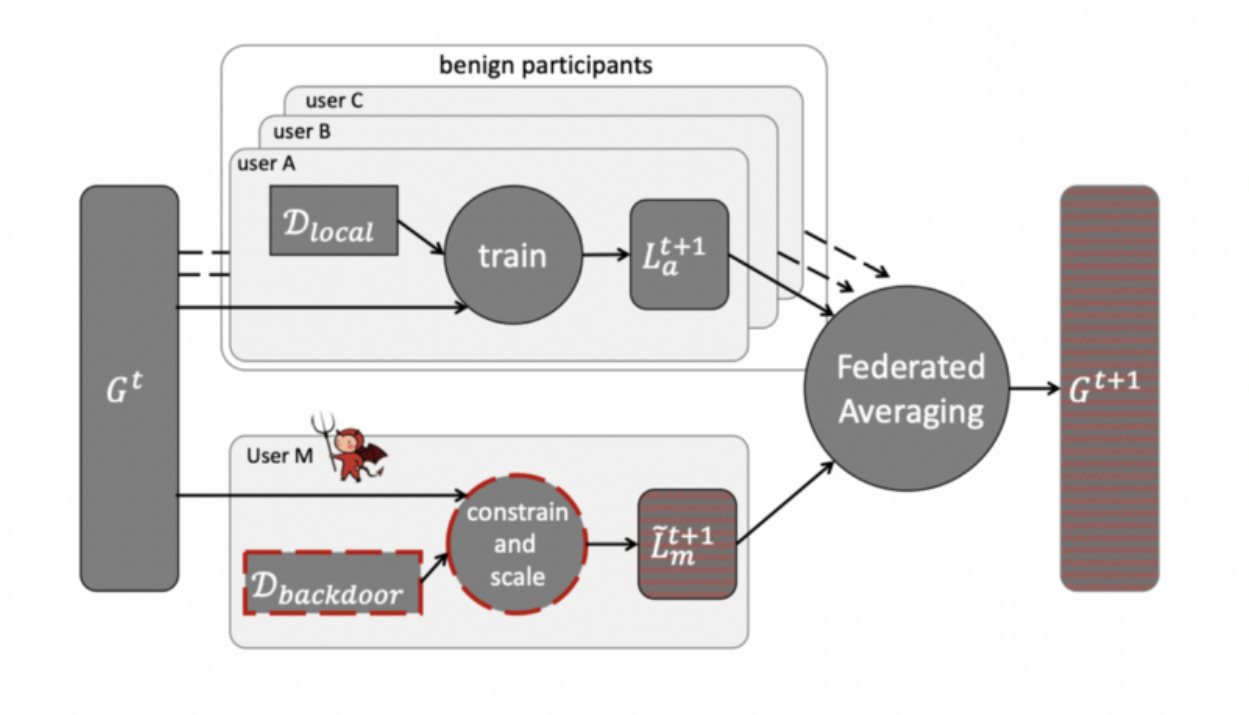
Data Poisoning

If the training data has clean-labels, the attack can't be effective for the malicious attacker since the attackers can't change the label for clean-labels. This is because the labels are certified and serialized, making it impossible to manipulate the labels.[1] However this doesn't stop the attacker from feeding the model dirty labels. Malicious attackers can instead introduce training data to the model which is misclassified.[7]

If the learning system doesn't support clean-labels, malicious attackers pose several different attacks on the data. The most common is **label-flipping**. In this attack, labels of good training data is changed to a different class. If you had a number image classifier, the classifier could be taught to infer that an image of the picture 1 is 7.[1] Another type of attack is **backdoor poisoning**, in this case the features are modified to blunder the predictions of the model. Since these attacks require access to the user's device, it requires sophisticated technical skills as well as computational resources making the attack quite challenging (but not impossible).

Model Poisoning

Instead of touching the data, here the malicious attacker will try to attack the updates of the local model on the user's device.[1] This leads to the model to misclassify with high confidence.[8] One of the common techniques of conducting this attack is by gaining access to a user's device. Instead of training the model on the local data, the attacker will provide a separate poisoned dataset with which local model will train with. This would then send these updates back to the global model.[7] The process sounds similar to data poisoning, except here you aren't really manipulating training data of the user. Similar to data poisoning, the process is challenging, but not impossible.

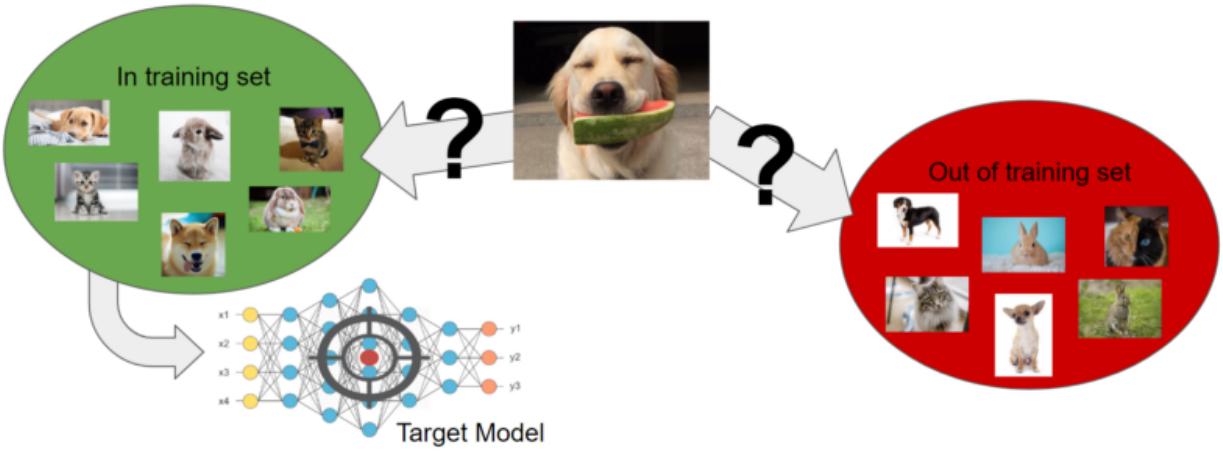


Model poisoning in action[8]

How Effective Are These Attacks?

Research has shown that model poisoning attacks are far more effective as it allows the attacker to be more evasive. They use stealth techniques, such as alternating minimization strategy[9], to do this and make the model misclassify. Most attackers will combine data and model poisoning attack since they both try to fundamentally attack the behavior of the model.[1]

2. What Are Inference Attacks?



A type of inference attack where the user can determine if a data point was used in the training phase[13]

So we talked about how these malicious attackers can tamper with the data and model to manipulate the predictions. But what information can attackers actually get about a user from the model during the training phase (we are still talking about vulnerabilities in the **training phase**)?

Inference attacks typically occur with gradients (this is used in deep learning algorithms). Remember, the gradients these models are using are essential to update the model and improve its accuracy. Without gradients, we can't create these complex models. However, malicious attackers during the model update phase this can extract significant amount of information such as class representative, membership, properties and in some cases the original training data.[1]

Inferring Class Representatives

This attack allows malicious participants to create a Generative Adversarial Network (GAN) attack on the model. The attacker can train a GAN to essentially generate similar samples of targeted training data which were originally intended to be private.[1] It's not the original data, but the distribution is very similar to the training data. This attack may not seem so bad since the model is trained on several different user's data. This GAN attack would prove to be effective had all the data the model been training on come from one user.[1] This is unlikely, but regardless invokes a privacy leak in some regard.

Inferring Membership and Properties

If a malicious attacker has a data point, they can use it to figure out if it was used by the model during the training process.[1] For example if you had a patient profile, you could check if it were used in the the training model. Here the malicious attacker can be either passive or active. In the passive scenario, the attacker will just observe for his/her amusement. But in the active scenario, the attacker will aggressively try to obtain even more information about the user by sending malicious updates to the model.[1]

Inferring Training Input And Label

Lastly, Deep Leakage From Gradients[10] and Improved Deep Leakage From Gradients[11] both present algorithms that can obtain the original training dataset from the model. With just a couple iterations, the algorithm can generate the original image that was used to train the model.[1] This is far more powerful than the previous attacks we observed since those were either creating a general distribution of the original data or extracting additional information.

So What Is The Future?

There are several ideas and suggestions to improve the current model for the future. These ideas include:[1]

- Whether we even need to share model updates
- Try to implement new or with existing architectures
- Share less sensitive information and provide only predictions
- Decentralizing federated learning
- Differential privacy
- Check network for malicious attackers

Given all this, federated learning is on the right track to provide privacy and accurate models. Malicious attackers will always try to find their way into the system. But research and studies are being done to improve and protect federated learning from these attacks.

References

[1] https://arxiv.org/pdf/2003.02133.pdf (original paper this article is based on)

[2] https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[3] https://stock.adobe.com/images/layered-paper-cut-out-colored-paper-human-profile-with-brain-and-man-stealing-ideas-concept-of-stealing-idea-intellectual-property-copyright-origami-paper-art-style/207838383

[4] https://en.wikipedia.org/wiki/Inference_attack

[5] https://community.arm.com/developer/ip-products/processors/b/ml-ip-blog/posts/ai-basics-training-vs-inference-whats-the-difference

[6] https://blogs.nvidia.com/blog/2016/08/22/difference-deep-learning-training-inference-ai/

[7] https://www.princeton.edu/~pmittal/publications/bhagoji-icml19.pdf

[8] https://arxiv.org/pdf/1807.00459.pdf

[9]https://arxiv.org/pdf/1811.12470.pdf

[10] https://arxiv.org/abs/1906.08935

[11] https://arxiv.org/abs/2001.02610

[12] https://www.inovex.de/blog/federated-learning-frameworks-part-2/

[13] https://gab41.lab41.org/membership-inference-attacks-on-neural-networks-c9dee3db67da