HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 1 (2023)

Double Momentum Backdoor Attack in Federated Learning

Satwik Panigrahi[1], Nader Bouacida[2] and Prasant Mohapatra[2#]

[1]Dougherty Valley High School
[2]University of California, Davis
[#]Advisor

ABSTRACT

Federated learning is conceived as a privacy preserving framework that trains deep neural networks from decentralized data. However, its decentralized nature exposes new attack surfaces. The privacy guarantees of federated learning prevent us from inspecting local data and training pipelines. These restrictions rule out many common defenses against poisoning attacks, such as data sanitization and traditional anomaly detection methods. The most devastating attacks are usually the ones that corrupt the model without altering the performance of the main task. Backdoor attacks are prominent examples of adversarial attacks that often go unnoticed in the absence of sophisticated defenses. This paper sheds light on backdoor attacks in federated learning, where we aim to manipulate the global model to misclassify the samples belonging to a particular task while also maintaining high accuracy on the main objective. Unlike existing works, we adopted a novel approach that directly manipulates the gradients' momentums to introduce the backdoor. Specifically, the double momentum backdoor attack computes two momentums separately based on malicious and original inputs and uses them to update the model. Via experimental evaluation, we demonstrate that our attack scenario is capable of introducing the backdoor while successfully evading detection.

Introduction

The maturation of deep learning as a practical approach for Machine Learning (ML) and the explosive expansion of big data have enabled many modern breakthroughs in Artificial Intelligence (AI). With a wide range of deep neural networks, affordable and accessible computing power becomes a cornerstone for developing ML models. Besides, the datasets used in deep learning are usually crowdsourced and may contain profoundly sensitive information [1]. This factor, compounded by the fact that the data can be misused or leaked, is the reason why many regulatory bodies have taken measures to enact stricter privacy protection laws [2]. The recent proliferation of smart devices with unprecedented processing power [3] and the stricter data protection laws have given rise to federated learning [4] – [8]. Federated learning has emerged as a revolutionary paradigm for massively distributed training of deep learning models with thousands and even millions of mobile devices [9]. It distributes the learning process to the edge in a sequence of training rounds. In each round, the central server broadcasts the current global model to a random subset of the clients. Each participant trains a local update using the global model and sends the newly computed update back to the server. The latter aggregates the updates into a new global model and restarts the training process following the same steps. In each iteration, the shared global model is improved until reaching convergence. By conducting model training at the network edge, federated learning prevents the server from accessing clients' local data or training pipelines [10]. Despite that federated learning revolves around privacy, ownership, and locality of the data, its decentralized concept results in new vulnerabilities [10] – [12]. Indeed, federated learning is by design vulnerable to model

poisoning [13], [14]. Adversaries exploit the fact that they directly control the data and the local training procedure. Hence, they can contribute with corrupted updates as part of the decentralized training. Sharing model parameters with each participant opens numerous avenues to exploit the risks associated with the federated learning environment. When this happens, compromised clients can conspire to simultaneously submit malicious model updates that are trained on both benign and malicious data samples to mislead the global model. The security of federated learning itself is critical to design trustworthy collaborative training frameworks. The privacy-preserving and decentralization promises of federated learning have attracted different applications where data is sensitive and challenging to collect in a central entity. However, federated learning cannot guarantee that all participants are honest by relying solely on their security configurations. Furthermore, many of the existing defenses [15], [16] against adversarial attacks can be off limit for federated learning since they require a careful inspection of training data or a complete control of the training process. This paper focuses on backdoor attacks, a prevalent category of adversarial attacks in federated settings. Under backdoor attacks (also known as targeted attacks), the adversary aims to modify how the model behaves on specifically selected sub-tasks while preserving good overall performance on the main task, e.g., the attacker can trick an image classifier so that it assigns attacker-chosen labels to images belonging to a particular class. Backdoor attacks [17]– [21] are less transparent and harder to detect in comparison to untargeted attacks. Targeted attacks on federated learning can be tricky to achieve, given the inherently heterogeneous and unbalanced data distribution across clients. In this paper, we propose the double momentum backdoor attack that introduces a backdoor using two momentums calculated respectively based on malicious and original inputs. Computing a separate momentum for the backdoor data and the client's intact dataset ensures the stability and stealthiness of the attack. If the backdoor is successfully embedded into the model updates using a combination of benign and malicious momentums, compromised clients in federated settings can directly influence the weights of the global model and train in any optimization direction that benefits the attacker. To preserve the attack from detection, we redesign the loss function to consider the possibility of diverging from what the aggregator considers within the "norm", so that the attack can unfold without detecting suspicious behavior. To validate the effectiveness of our new attack, we craft poisoned data samples as backdoor data and set up different scenarios. The experimental results show that the double momentum backdoor attack launched by a single attacker can successfully inject the backdoor into the global model. We summarize our main contributions and findings as follows:

- We introduce the double momentum backdoor attack and demonstrate the performance of this attack on two concrete learning tasks: handwritten digits recognition on the EMNIST dataset and image classification on the CIFAR-100 dataset. We show that our method is more effective and persistent than traditional backdoor attacks.
- We incorporate the evasion of common defenses against backdoor attack into loss function, so the poisoned model updates look and behave similarly to models trained without backdoors.
- We evaluate this new attack resiliency against the state of-the-art defenses.

Related Work

A growing body of literature has examined backdoor attacks on federated learning. In their cutting-edge paper, Bagdasaryan et al. [18] proposed a model replacement approach that introduces backdoor functionality into the shared model by constraining and scaling up the attacker's updates. Bhagoji et al. [22] considers the case where the adversarial objective is to cause the model to misclassify a set of chosen inputs with high confidence. To carry this type of attack, they explored the boosting of the malicious client's update to overcome the effects of benign updates. Besides, they proposed an alternating minimization policy to enhance attack stealth, which alternately optimizes for the training loss and the adversarial goal. Authors in [17] focused on model update poisoning attacks that allow non-malicious clients to have intact data samples from the backdoor tasks. In this scenario, the attacker trains a malicious model based on a benign dataset and a malicious dataset that describes the backdoor task. The cited works concentrate on semantic backdoors, which cause the backdoored model to produce adversary-chosen outputs on intact digital inputs.

For example, a backdoored image classification model predicts the images with specific features to belong to an attacker-chosen class, e.g., all images that contain green cars will be misclassified as birds. Other works on backdoor attacks [19], [21] consider trigger pattern backdoors. This type of backdoor attack requires the adversary to modify a subset of pixels for the trained model to misclassify the altered image. The trigger-pattern can be decomposed into separate local patterns across multiple parties as demonstrated in [19]. Our scheme focuses on the more powerful semantic backdoors.

Threat Model

Adversary Capabilities

In federated learning, an attacker can get full control over one or several clients, e.g., client devices whose application software has been compromised by malware. Besides, the adversary can stimulate multiple fake clients to carry out more successful attacks. In this paper, we define the capabilities of an adversary as follows

- The adversary controls the training data of any compromised device.
- The adversary controls the local training procedures, including the optimizer and the hyperparameters.
- The adversary can modify or replace the local model update before submitting it to the server.
- The attacker can dynamically adjust its local training algorithm or settings from one training round to another.

On the one hand, the adversary has no control over the aggregation algorithm used to average clients' updates at the server, nor any features related to the benign clients or the server. On the other hand, the central server cannot check the clients' datasets or set rules for local training. This threat model is common in federated learning applications because it exposes the vulnerabilities induced by decentralized training frameworks.

Attack Objectives

The adversary aims to insert a hidden backdoor into the global model while retaining the accuracy of the main task. The attack will produce a trained model that achieves high accuracy on both the chosen backdoor subtask and the main task when reaching convergence. Moreover, the trained model should be able to retain a good performance on the backdoor task for several training rounds after its insertion. The adversary will employ model replacement to submit the poisoned model to the server. For example, a successful attack on a digit recognition task produces a trained model that misclassifies 7s and predicts them to be 1s from multiple target clients (backdoor subtask) while correctly classifying other digits. The backdoored model behaves according to the adversary's objective. However, the performance of the poisoned model on correct inputs should not be affected.

Untargeted adversarial attacks exploit the model class representations boundaries to produce wrong predictions. By contrast, backdoor attacks intentionally stir these boundaries in a direction where specific inputs are wrongly classified. We dedicate this work to tackle semantic backdoors because they do not require modified inputs by the attacker. Moreover, semantic backdoors can represent a more significant threat to federated learning than trigger-pattern backdoors, especially in applications where the data is outsourced from live environment interaction such as self-driving cars.

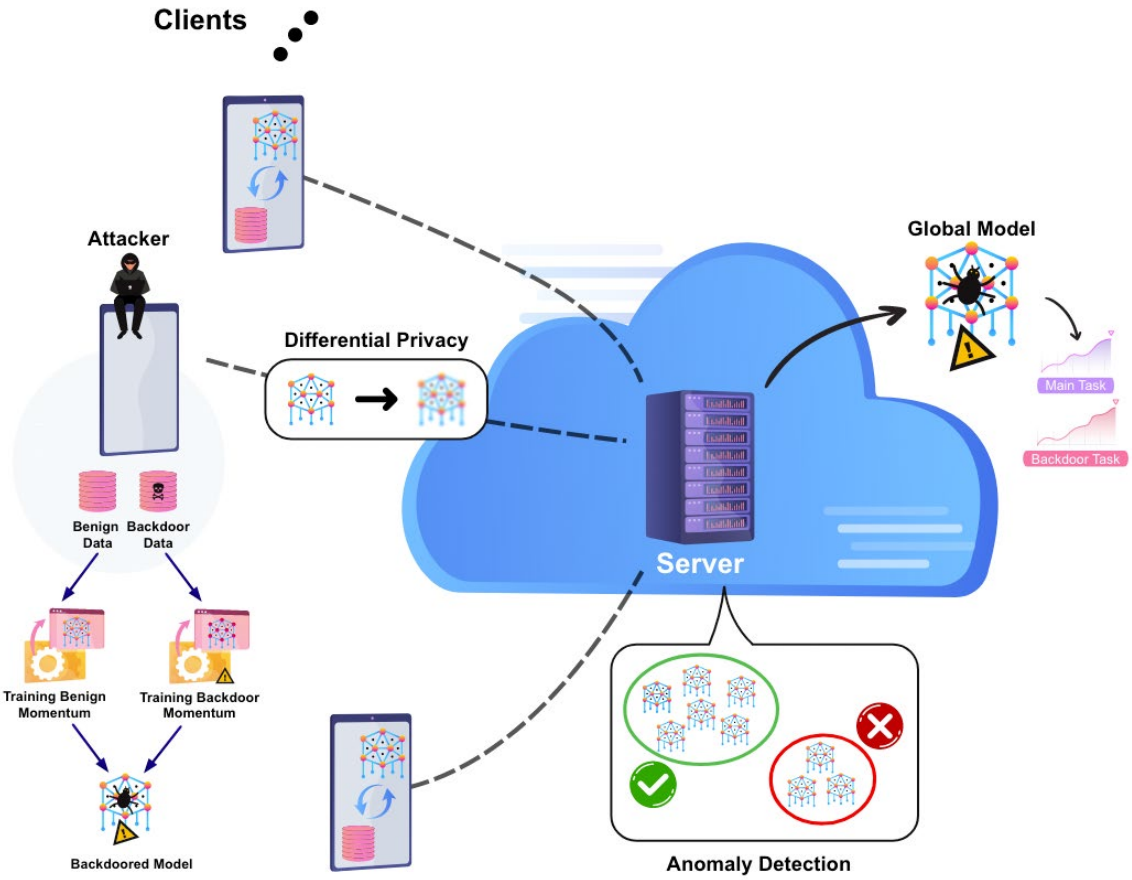

Figure 1. Overview of the double backdoor attack in federated learning.

Constructing the Backdoor Attack Model

Naive Approach and Baseline

The adversary can train its model on backdoored samples. In the same way as [23], each training batch should contain a mix of correctly labeled data and backdoored data to assist the model in learning to notice the difference. The adversary can also adjust the training parameters to escalate the overfitting to the backdoored inputs. Federated learning applies malicious clients' model updates directly to the global model using the aggregation algorithm, thus injecting the backdoor. However, aggregation can cancel out most of the corrupted models' contributions, and the backdoor will be quickly forgotten. There is no guarantee that the attacker will be selected frequently enough for training to sustain the attack effect. Besides, such a naive attack can take many rounds to achieve reasonable performance. In our experiments, we use the more advanced backdoor scenario in [17] as a baseline for comparison, where the backdoored model is trained on a benign dataset and a malicious dataset coupled with a boost factor.

Double Momentum Approach

Federated learning distributes the training of a deep learning model across N clients by iteratively aggregating local model updates into a shared global model. There exist many flavors of distributed learning. We focus on synchronous federated learning, which proceeds in rounds of training. It aims to learn a global model with parameters embodied in

a real tensor W from data stored across a number of clients. In training round $t \geq 1$, the server distributes the current global model W_t to a subset S_t of K selected clients where $K < N$ is the number of clients per round. The selected clients locally train the model W_t based on their data and independently update it. As the result, they produce new local models $\{W_t^k \mid k \in S_t\}$ and send the difference $\Delta W_t^k = W_t^k - W_t$ (usually referred to as the model update) back to the central server. The server updates the joint global by aggregating ΔW_t^k as follows:

$$W_{t+1} = W_t + \eta \frac{\sum_{k \in S_t} n_k \Delta W_t^k}{\sum_{k \in S_t} n_k}$$

where η is the server learning rate and n_k is number of data instances in client k. This process will be iterated until the global model reaches convergence.

Our proposed backdoor scenario changes how the local models are trained to insert a backdoor using a double momentum. Fig. 1 illustrates an overview of the attack model. To craft a backdoored model W^*, we assume that the attacker has a mix of malicious backdoor samples D_{mal} (data instances with attacker-chosen labels) and a set of benign training samples generated from the true distribution D_{true} of the dataset. As we will show in this work, a single attacker can submit a backdoored model W^* which is not only trained on the backdoor data but also carries out the backdoor functionality to the global model.

When an attacker $k \in S_t$ is selected in round t, double momentum backdoor attack works on two stages. The first stage is to train the global model received from the server W_t using Stochastic Gradient Descent (SGD) with Momentum exclusively on intact data belonging to the true distribution D_{true} to generate the first momentum U_t^k as follows:

$$U_t^k = \boldsymbol{m}. U_{t-1}^k - \mathrm{lr}. \nabla_{W_t} L_U(W_t, X^k, y^k)$$

where m is a hyperparameter of the momentum, which takes values between zero and one, lr is the client learning rate, and L_U is the loss function for the first momentum. X^k and y^k are respectively intact data instances and labels sampled from the original dataset D_{true}.

The second stage of backdoor insertion involves training the joint model W_t using Momentum SGD exclusively on backdoor data (X_b, y_b) to produce the second momentum V_t^k as follows:

$$V_t^k = m \,. V_{t-1}^k - lr. \nabla_{W_t} L_V(W_t, X_b, y_b)$$

where X_b and y_b are respectively the data and labels of the malicious samples from the backdoor data D_{mal} and L_V is the loss function for the second momentum. For the sake of simplicity, we used the same learning rate lr and momentum parameter m. Also, we employ the same backdoor dataset for all compromised clients. However, it should not be considered as a general rule. Then, we combine both momentums using a weighted sum and compute the model update of the client k as follows:

$$W^* = W_t + \alpha U_t^k + (1 - \alpha) V_t^k$$

where $0 < \alpha < 1$ is a hyperparameter of our backdoor attack which we designate as the double momentum scaling factor. The weighted sum relates to the dynamics of double momentum. We create a new velocity variable to store each momentum for both intact and backdoored data. The weights allow us to control the fragile balance between backdoor insertion and discretion. They also ensure that the attack is not aggressively pushing the local model's weights into the backdoor's direction, resulting in smooth poisoning of the model. In the remainder of this section, we refer to the first momentum U_t^k as the benign momentum and the second momentum V_t^k as the malicious momentum the backdoor's direction, resulting in smooth poisoning of the model. In the remainder of this section, we refer to the first momentum U_t^k as the benign momentum and the second momentum V_t^k as the malicious momentum.

Finally, the adversary ambitiously attempts to substitute the whole model by the backdoored model W^* by transmitting:

$$\Delta W_t^k = \beta(W^* - W_t)$$

$\beta = \frac{\sum_{i \in S_t} n_i}{\eta n_k}$ is a boost factor that ensures the attacker's backdoor contribution survives the aggregation and impacts the global model. If multiple malicious clients appear in the same round, they coordinate with each other to divide this update evenly. An attacker may not know the values of the server learning rate η and the number of data samples in other clients to define the booting factor β. In this case, he can gradually increase the boosting factor β going through each round in a probing fashion and measure the backdoor task's accuracy.

Since the learning rate is fixed, the standard gradient descent method will converge slower and sometimes even fall into a local optimum. Momentum SGD improves the stability of the learning process and the endurance of the attack because historical gradients will lead the parameters to converge faster towards the optimal value. Computing a separate momentum for the backdoor data and the user's original data allows us to discretely and smoothly insert the backdoor. Boosting the impact of the backdoored model update guarantees that the introduced backdoor survives the aggregation, and the global model is severely contaminated. Furthermore, this effect harnessed with the heuristic of the exponential moving average of the gradients calculated based on the backdoor samples ensures that the attacker's contribution is smoothly transferred to the global model. In fact, if the current gradient descends in the same direction as the last update, it can positively accelerate the current search to optimize the weights for the backdoor task performance. Conversely, the benign momentum can act as deceleration to the current search. Heightening the backdoor impact serves in any round of federated learning but is more useful when the global model is close to the convergence stage. Following the attack, the poisoned global model should exhibit high accuracy on the backdoor task without affecting the main task.

Defense Evasion

Our attack is effectively a two-task learning scheme, where the model learns the main task using the first momentum U_t^k and the backdoor subtask using the second momentum V_t^k. The goal is to maintain high accuracy for both tasks after introducing the backdoor. In this section, we demonstrate the techniques that enable the adversary to generate backdoored models that look legitimate. We enhance the double momentum backdoor attack with sophisticated defense evasion methods. These strategies allow the production of a resilient backdoored model that scores high accuracy on both the main and target tasks yet is not dismissed by the server's anomaly detector or diluted by artificial noise.

A. Evading Anomaly Detection

Anomaly detection is considered a more proactive type of defense that explicitly detects malicious updates and blocks their impact on the system. In federated learning environments, attacks such as data poisoning and model poisoning can be discovered using anomaly detection techniques.

Since the central server has no access to the training data at the clients, most popular anomaly detection mechanisms [14], [24]–[26] in federated learning try to identify abnormal model updates and discard them. When the algorithm trains the global model on the backdoor data, the generated malicious momentum V_t^k will most likely cause the backdoored model W^* to significantly diverge from the global model and other benign local models. In this case, the backdoored model update can be spotted by an advanced anomaly detector and ignored as a result. Evading anomaly detection is critical to the attack's success.

By using double momentum backdoor, we can control the impact of the malicious samples on the poisoned model. Unlike some exciting works, double momentum backdoor attack does not make a sudden change to the model weights but instead creates a gradual shift towards the successful backdoor insertion. While this smooth backdoor

injection will help the attack go undetected, we want more assurance concerning the attack's success in the presence of anomaly detection. Thus, we incorporate defense evasion into the training using a particular loss function that penalizes the model for deviating too much from the benign model. To be specific, we modify the loss function L_V of the malicious momentum by adding a term $p.\left(1 - Cosine(\Delta_U W_t^k, \Delta_V W_t^k)\right)$, which represents the cosine similarity between the model updates generated for both benign and malicious data samples separately using the standard loss function (p is a distance factor).

Unconstrained backdoor attacks can be defended by norm thresholding of the model updates [17]. Since our attack is boosted, it is more likely to produce model updates with a large norm. A common defense is for the aggregator to reject model updates whose norm exceeding a pre-defined threshold M. To overcome this type of defense, we can also consider bounding the model update by M after being boosted by a factor of β. This can be done by projecting the locally trained model into the l_2 ball of size M/β around W_t. We can assume that the attacker does know the threshold M of the norm clipping defense deployed in the server or employ a probing mechanism that estimates the threshold M.

B. Evading Differential Privacy

Differential privacy [27] injects statistical Gaussian noise into the updates after norm clipping. The goal of differential privacy is to guarantee with high confidence that no single data record can be meaningfully distinguished from the rest. Traditionally, the quantity of noise added to obtain good differential privacy is relatively significant. Since our intent is not privacy by any means but instead diluting the impact of the backdoor attack, we apply a small amount of noise that is empirically sufficient to restrict the success of the attacks.

One possible avenue to evade differential privacy is applying a denoising filter to the model updates by scaling the gradients based on their value to the model utility before updating the global model. The utility is defined as the closeness between the model updates and their original values before adding differential privacy. Since the original values are private, we can approximate this utility with the distance from the noise distribution using region variance estimation [28]. This evasion process requires the attacker to have control over the server or aggregation algorithm, which is not the case in our threat model. We opt not to weaken the threat model and keep it as close to reality as possible by testing how well the differential privacy will affect our proposed attack.

Experimental Evaluation

A. Experimental Settings

1) Datasets: We conduct experiments on TensorFlow Federated [29], a benchmark for federated learning using two popular datasets: (1) EMNIST dataset [30] for handwritten digit recognition, which serves as a more complex extended version of the popular MNIST dataset. The data is grouped based on the writer of the character for a total of 3383 clients. (2) CIFAR-100 dataset [31] with coarse labels for 20class image classification (superclasses). The data partitioning across 500 clients is performed using a hierarchical Latent Dirichlet Allocation (LDA) process [32]. We opt for these datasets because they are suitable for our attack scenario (semantic backdoors), and we want to show the performance of the double momentum backdoor attack across different applications and scales.

2) Models: For FEMNIST's image classification task, we use a Convolutional Neural Network (CNN) with two 5x5 convolution layers, each of them followed with 2×2 maxpooling, a fully connected dense layer with 2048 units, and a final SoftMax output layer. For CIFAR-100 dataset, we consider a more complex VGG-16 architecture.

3) Sampling of Adversaries: Since our attack is supposed to inject the backdoor using a single shot, we carry out a fixed frequency attack scenario where a single adversary appears in every 1/f round. In the experiments, we consider decreasing the frequency of the attack to test if our attack scheme is able to sustain the backdoor inserted in the global model.

4) Backdoor tasks: Recall that the attacker's goal is to ensure that the model behaves differently on some targeted tasks in backdoor attacks. We allow benign clients to have nonmalicious data instances from the backdoor tasks. For instance, if the attacker desires the model to misclassify the digits 7s as 1s, we allow benign clients to have some intact samples correctly labeled as 7s. Besides, we construct the backdoor by collecting examples from multiple clients. Since samples from different clients follow different distributions, we obtain a diverse backdoor dataset.

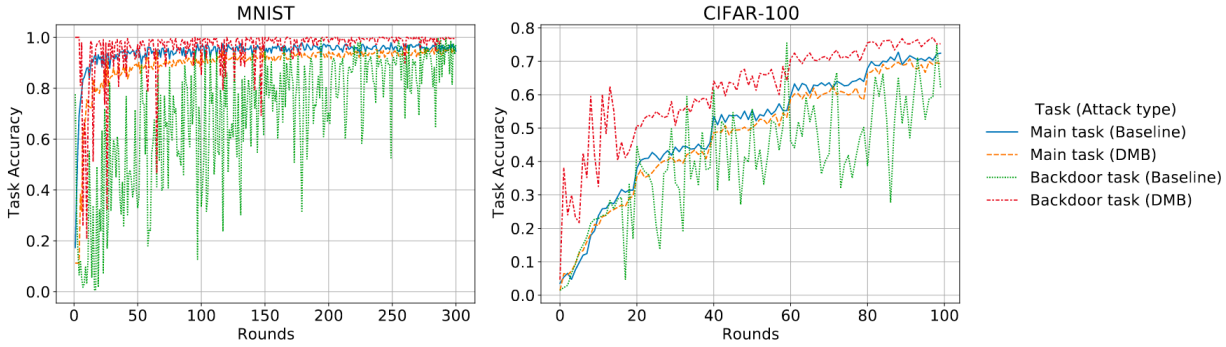

Figure 2. Fixed-frequency attack where a single attacker appears in every training round ($f = 1$).

TABLE I
SUMMARY OF THE EXPERIMENTAL SETTINGS.

	EMNIST	CIFAR-100
Client Learning Rate (lr)	0.1	0.15
Batch Size	16	64
Number of Local Epochs	5	5
Momentum (m)	0.9	0.9
Clients Number per Round	10	8
Sever Learning Rate(η)	1	1
Double Momentum Scaling Factor	0.4	0.5

B. Results

First of all, we conduct experiments on the EMNIST dataset. The latter is a handwritten digit image classification dataset where digits are gathered from 3383 users, each with around 100 images of digits and with its unique writing style. We train a shared global model in a federated learning fashion using the Federated Averaging (FedAvg) algorithm [9]. Table I summarizes the experimental settings1. We run all experiments for 300 rounds of federated learning. In this experiment, we consider the backdoor task of misclassifying the digits 7s from several targeted clients as 1s. The backdoor dataset is sampled from 20 different clients. We bound the malicious model updates using a norm bound of 0.33 before boosting, and we apply a small amount of Gaussian noise to the updates (differential privacy) with σ = 0.025 before submitting the model update to the server. At the server side, we activate the anomaly detection mechanism discussed in section V-A 2. Fig. 2 shows the results for the single-attacker scenario where a single adversary-controlled client is selected once in each round. The task accuracy of the backdoor task is the fraction of testing data

samples representing this task (samples that hold the digits 7s) and misclassified as desired by the attacker (misclassified as 1s). Meanwhile, the accuracy of the main task represents the conventional testing performance on other digits.

TABLE II
NUMERICAL RESULTS SUMMARY.

	EMNIST	CIFAR-100
Main Task Accuracy (Baseline)	94.6%	72.4%
Main Task Accuracy (DMB)	94.52%	69.2%
Average Backdoor Task Accuracy (Baseline)	69.2%	40.7%
Average Backdoor Task Accuracy (DMB)	95.8%	61.2%

Double Momentum Backdoor (DMB) attack succeeds in injecting the backdoor with high accuracy. As shown in Table II, the average backdoor task accuracy over 300 rounds of federated learning is 95.8%. It significantly outperforms the baseline attack scenario that only achieves 69.2% backdoor accuracy. The accuracy of the main task remains unaffected (94.52% compared to 94.6%). We draw the attention of the reader that the main task accuracy in Table II is measured at the end of the training process. Moreover, as illustrated in Fig. 2, our attack looks more stable than the baseline attack, given that our backdoor task accuracy oscillates less than the baseline. Under double momentum backdoor attack, we find a prominent phenomenon were using double momentum to inject the backdoor ensures the attack's stability and stealthiness. The high attack success rate also suggests that our defense evasion mechanisms are working as intended. Since the global model is smoothly moved in the backdoor direction, there is a high likelihood that it again converges to a model that includes the backdoor without sounding the alarms of the anomaly detection mechanism. We will explain later in this section why the safeguards in place failed to prevent the attack. For the second scenario of CIFAR-100 with coarse labels, we follow a similar approach. In this experiment, we consider the backdoor task of misclassifying the images belonging to the class "fish" as "aquatic mammals". CIFAR-100 image classification application is far more complicated than the EMNIST dataset, and backdoor insertion can be challenging. Fig. 2 clearly shows the superiority of the Double Momentum Backdoor (DMB) attack in comparison to the baseline. The performance gap is quite large: the average backdoor accuracy for DMB is 61.2% compared to only 40.7% for the baseline. The main task is almost untouched, with a negligible drop in accuracy. Despite the complicated VGG-16 model that we employed, the double momentum backdoor attack was able to prove again its capacity to inject backdoors. The combination of momentums ensures the successful backdoor poisoning throughout the training process.

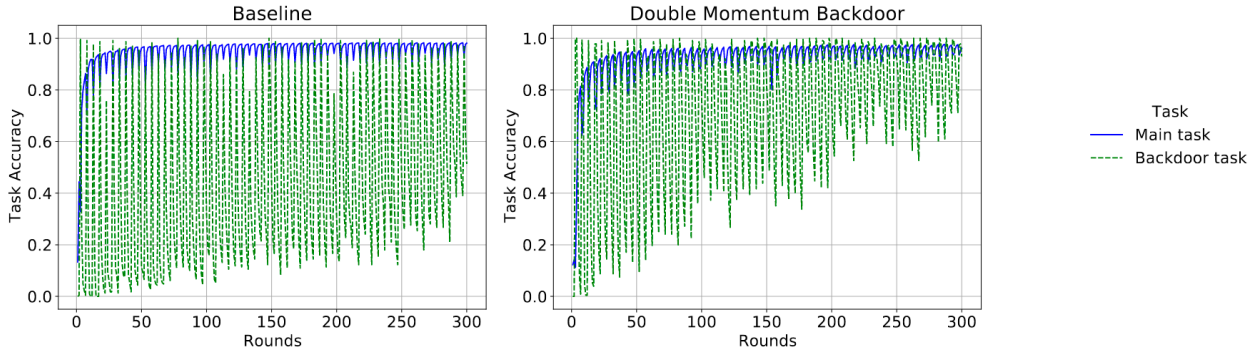

Figure 3. Attack on EMNIST dataset where a single attacker appears in every 5 training rounds ($f = 1/5$)

The results validate the fact that our scheme can mimic the normal updates while achieving its attack objective. The malicious updates generated by the attacker can easily pass for legitimate model updates and fool the anomaly detector and other federated learning safeguards. The baseline attack is easier to spot compared to our scheme. Our proposed attack can inject the crafted backdoor into the global model in a smooth and discrete manner and achieves significantly higher performance than the baseline.

Conclusion

Federated learning enables clients, some of whom may be potentially malicious, to manipulate the global model through their model updates submitted to the server. Adversaries can exploit vulnerabilities in federated learning by acting as legitimate clients to inject a backdoor into the shared model. In this paper, we develop a new backdoor attack that exploits these vulnerabilities, and we demonstrate its effectiveness on standard federated learning applications. Double momentum backdoor attack has proven its capability to contaminate the trained model effectively using only a single attacker in each training round. To remain stealthy against defense strategies, we incorporated a sophisticated loss function that penalizes the model for diverging from what the aggregator considers within the "norm". The proposed attack succeeded in defying defense proactive defense techniques by incorporating defense evasion techniques and adding more stability and smoothness in the way we introduce the backdoor.

In federated learning systems that operate at larger scales, it might be impractical to establish an enforceable collaborative agreement to monitor potentially malicious participants. Other than inserting backdoors, some compromised clients may purposely try to deteriorate performance, bring the system down, or extract sensitive information from other parties. Hence, security strategies will be required to mitigate these risks by putting an end-to-end security system in place, such as advanced encryption of model submissions, traceability of actions, verification systems, execution integrity, model confidentiality, and protections against suspect actions.

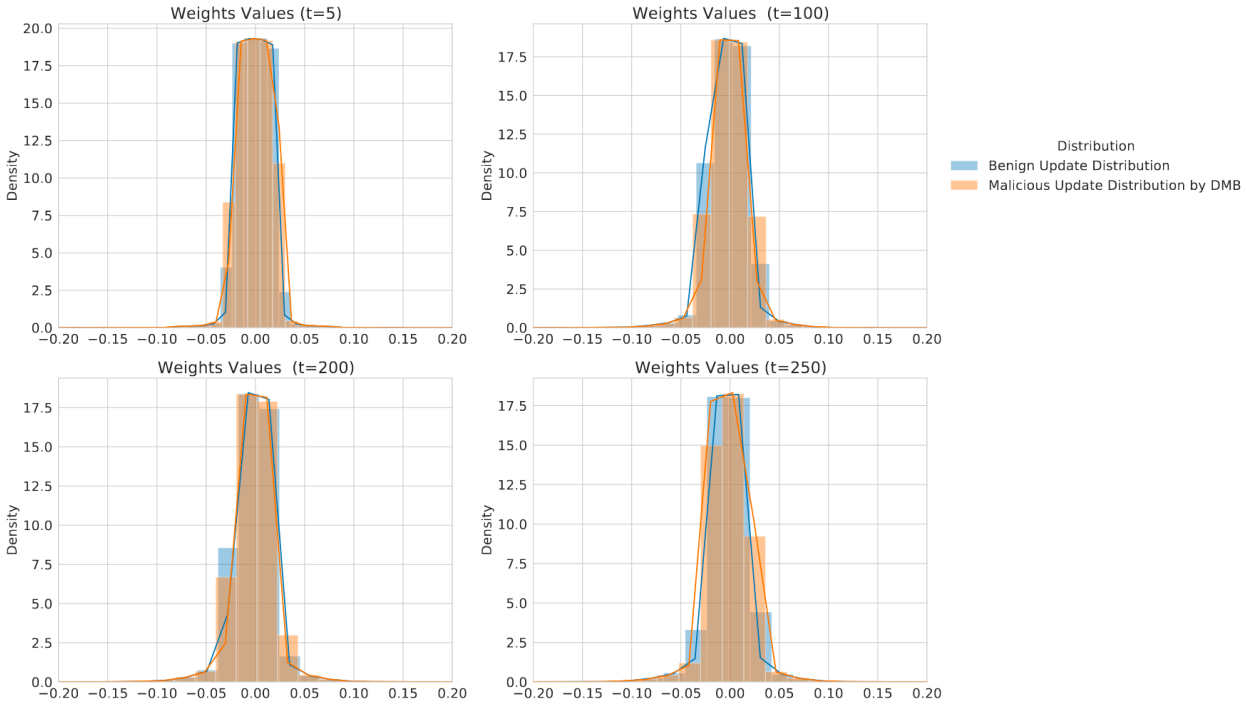

Figure 4. Comparison of model update weights distributions between normal updates and malicious updates generated by Double Momentum Backdoor (DMB) before boosting and differential privacy (on EMNIST Dataset).

References

[1] F. Mirshghallah, M. Taram, P. Vepakomma, A. Singh, R. Raskar, and H. Esmaeilzadeh, “Privacy in deep learning: A survey,” arXiv preprint arXiv:2004.12254, 2020.

[2] P. Voigt and A. v. d. Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, 1st ed. Cham, Switzerland: Springer Publishing Company, Incorporated, 2017.

[3] S. Nunez, “Your phone is now more powerful than your PC,” August 2020. [Online]. Available: https://insights.samsung.com/2020/ 08/07/your-phone-is-now-more-powerful-than-your-pc-2

[4] J. Koneˇ cn ´y, H. B. McMahan, D. Ramage, and P. Richt ´arik, “Federated Optimization: Distributed machine learning for on-device intelligence,” arXiv preprint arXiv:1610.02527, Oct. 2016.

[5] J. Koneˇ cn ´y, H. B. McMahan, F. X. Yu, P. Richtarik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” in NIPS Workshop on Private Multi-Party Machine Learning, 2016.

[6] 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, vol. 54, 2017, pp. 1273–1282.

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” ACM Trans. Intell. Syst. Technol., vol. 10, no. 2, Jan. 2019.

[8] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, “A survey on federated learning: The journey from centralized to distributed on-site learning and beyond,” IEEE Internet of Things Journal, 2020.

[9] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Koneˇ cn ` y, S. Mazzocchi, H. B. McMahan et al., “Towards federated learning at scale: System design,” arXiv preprint arXiv:1902.01046, 2019.

[10] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, “A survey on security and privacy of federated learning,” Future Generation Computer Systems, vol. 115, pp. 619–640, 2021.

[11] L. Lyu, H. Yu, and Q. Yang, “Threats to federated learning: A survey,” arXiv preprint arXiv:2003.02133, 2020.

[12] N. Bouacida and P. Mohapatra, “Vulnerabilities in federated learning,” IEEE Access, vol. 9, pp. 63 229–63 249, 2021.

[13] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Model poisoning attacks in federated learning,” in Proc. Workshop Secur. Mach. Learn.(SecML) 32nd Conf. Neural Inf. Process. Syst.(NeurIPS), 2018.

[14] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to byzantine-robust federated learning,” in 29th USENIX Security Symposium (USENIX Security 20), Aug. 2020, pp. 1605–1622.

[15] J. Steinhardt, P. W. Koh, and P. Liang, “Certified defenses for data poisoning attacks,” 2017, p. 3520–3532.

[16] K. Liu, B. Dolan-Gavitt, and S. Garg, “Fine-Pruning: Defending against backdooring attacks on deep neural networks,” in Research in Attacks, Intrusions, and Defenses, 2018, pp. 273–294.

[17] A. T. Suresh, B. McMahan, P. Kairouz, and Z. Sun, “Can you really backdoor federated learning?” arXiv preprint arXiv:1911.07963, 2019.

[18] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, vol. 108, 26–28 Aug 2020, pp. 2938–2948.

[19] C. Xie, K. Huang, P.-Y. Chen, and B. Li, “DBA: Distributed backdoor attacks against federated learning,” in International Conference on Learning Representations, 2020.

[20] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, “Attack of the tails: Yes, you really can backdoor federated learning,” arXiv preprint arXiv:2007.05084, 2020.

[21] A. Huang, “Dynamic backdoor attacks against federated learning,” arXiv preprint arXiv:2011.07429, 2020.

[22] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in Proceedings of the 36th International Conference on Machine Learning, vol. 97, 09–15 Jun 2019, pp. 634–643

[23] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," arXiv preprint arXiv:1708.06733, 2019. [Online]. Available: https://arxiv.org/abs/1708.06733

[24] 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, 2016, p. 508–519. [25] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018, pp. 19–35.

[26] S. Li, Y. Cheng, Y. Liu, W. Wang, and T. Chen, "Abnormal client behavior detection in federated learning," arXiv preprint arXiv:1910.09933, 2019.

[27] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, p. 308–318.

[28] L. V. Hedges, E. Tipton, and M. C. Johnson, "Robust variance estimation in meta-regression with dependent effect size estimates," Research synthesis methods, vol. 1, no. 1, pp. 39–65, 2010.

[29] "Tensorflow federated." [Online]. Available: https://www.tensorflow.org/ federated

[30] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," arXiv preprint arXiv:1702.05373, 2017.

[31] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," 2009.

[32] W. Li and A. McCallum, "Pachinko allocation: DAG-structured mixture models of topic correlations," in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 577–584.