Efficient Backdoor Defense for Federated Learning with Partial Model Inspection

Kyle Wilkerson, Palden Lama,Andrew Merrow,Ryan Bonnet,Vasudha Vedula,Rajendra Boppana
Department of Computer Science
University of Texas at San Antonio
kyle.wilkerson,andrew.merrow,ryan.bonnet,vasudha.vedula@my.utsa.edu;
palden.lama,rajendra.boppana@utsa.edu

***Abstract*—Federated Learning (FL) enables multiple agents to collaboratively train a global model without sharing local data, preserving privacy. However, this distributed setting introduces vulnerabilities to backdoor attacks, where malicious agents can poison the model by embedding trigger patterns into a subset of training data. Existing defenses such as robust aggregation, adversarial training, or pruning-based techniques either incur high computational overhead, degrade model accuracy, or fail under non-i.i.d. data distributions. Their effectiveness also diminishes for lightweight models.**

In this paper, we propose a lightweight and effective defense strategy that detects and mitigates backdoor attacks via partial model inspection. By analyzing the bias updates of the final layer, our method identifies statistical outliers indicative of malicious behavior. This approach reduces both computational cost and privacy risk, as it inspects only a small part of model updates. Experimental results show that our method significantly outperforms existing defenses such as MKrum, RLR, and Lockdown. When combined with RLR, it achieves robust defense under varying attack intensities in both i.i.d. and non-i.i.d. settings.

I. Introduction

Federated Learning (FL) is a privacy-preserving machine learning approach where multiple devices (referred to as clients or agents) collaboratively train a model under the coordination of a central server, while keeping the data decentralized. Instead of sending training data to a central server, each agent processes its data locally and sends only the model updates (e.g., learned weights and biases of neural network) to the FL server. The server aggregates these updates to improve the global model. This decentralized approach is valuable in privacy-sensitive or bandwidth-constrained computer vision applications, such as surveillance and healthcare [1], [2].

A critical vulnerability of FL stems from its privacy-preserving feature, which inadvertently creates a potential attack surface leading to model poisoning attacks. A malicious FL agent (attacker) can alter its training data to embed a hidden vulnerability ($i.e.$ backdoor) into the global model. This backdoor allows the attacker to manipulate the model by providing a specific input that triggers the model to behave in a way that is beneficial to the attacker [3]–[6]. These attacks remain elusive since the FL server does not have access to agents' training data, the backdoors may trigger only with specific input conditions, and the malicious agents may manipulate their model updates to resemble those of benign agents.

Traditional defense techniques using Byzantine-robust aggregation rules such as Krum [7], and coordinate-wise median [8] address untargeted model poisoning attacks that aim to reduce the model accuracy indiscriminately. However, they are not effective against targeted backdoor attacks. Recent works defensively adjust the server's learning rate based on agents' model updates [9] or detect malicious agents based on behavioral discrepancies among agents' models [10]–[12]. However, these techniques are less effective when a substantial number of FL agents are malicious or the agents' local datasets are not independent and identically distributed (non-i.i.d). Furthermore, many defense techniques [11]–[13] require the FL server to have unrestricted access to each agent's model updates, which increases the risk of privacy leakage [14].

Backdoor defense techniques, such as certified robustness [15] and adversarial training [16], incur high computational overhead. For example, certified robustness requires multiple inference passes, while adversarial training involves costly inner-loop optimization. These inefficiencies hinder their deployment in real-world resource-constrained FL settings. A recent pruning-based defense, Lockdown [17], uses isolated subspace training to decouple and prune suspicious parameters based on client consensus. Although effective, its performance degrades significantly for lightweight models.

In this paper, we present an efficient and robust method for detecting and mitigating backdoor attacks in FL. Our approach enables the FL server to perform anomaly detection on client updates through partial inspection, focusing on select model parameters to identify malicious behavior. We demonstrate that analyzing model updates at a fine granularity, such as the weights and biases of an individual neural network layer, reveals distinguishing features of malicious agents. In particular,

we detect outliers among the bias updates in the last layer of the neural network. Agents predicted as malicious during an FL training round are excluded from aggregation by the FL server. Our approach reduces both the computational burden on the FL server and the risk of privacy leakage, as only partial information from model updates needs to be inspected.

Experiments in the i.i.d. setting show that our approach outperforms existing defenses including MKrum [7], robust learning rate (RLR) [9], and Lockdown [17] by effectively mitigating backdoor attacks without compromising accuracy on clean validation data. In the non-i.i.d. setting, a hybrid strategy that integrates our method with RLR achieves substantially greater effectiveness than RLR alone. Notably, the hybrid approach delivers consistently robust performance across both i.i.d. and non-i.i.d. scenarios, even when a large fraction of FL clients are malicious. Further analysis shows that our method introduces minimal computational overhead, making it well-suited for deployment in resource-constrained FL environments.

The rest of the paper is organized as follows. Section II discusses the related works. Section III presents our defense technique. Section IV describes the experimental setup and Section V shows the evaluation results. Finally, Section VI concludes the paper.

II. Related Work

A. Backdoor Attacks

FL model poisoning attacks can be targeted or untargeted. Untargeted attacks, which aim to make the model converge to a sub-optimal minimum or completely diverge, are often detectable by observing the model's accuracy on validation data [7]. On the other hand, targeted or backdoor attacks aim to cause misclassification of specific inputs while preserving the model's overall performance [3], [4], [18]. For instance, an adversary might insert a small sticker, such as a cartoon icon, onto traffic sign images and label them as speed limits. At inference time, presenting a stop sign with the same sticker triggers the backdoor, causing it to be misclassified as a speed limit sign. Because the model continues to classify clean images correctly, the attack remains undetected on standard validation datasets.

B. Existing Defense Techniques

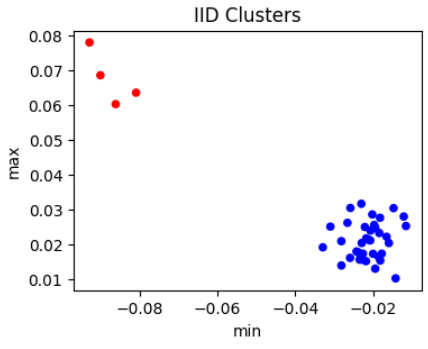
Traditional defense techniques against FL model poisoning attacks predominantly rely on robust aggregation methods such as Krum [7], coordinate-wise median [8] etc. Sun et al. [5] proposed to clip an agent's model update if its L2 norm exceeds a specified threshold, by dividing it with an appropriate scalar. The server then aggregates clipped updates and adds Gaussian noise to the aggregation. These methods can limit the extent to which modifications in global model parameters are influenced by malicious agents in the case of untargeted attacks. However, they are ineffective against backdoor attacks [3], [9], [19].

FLTrust [20] improves robustness by assigning trust scores to clients based on their alignment with a reference update derived from a clean validation dataset. However, this approach assumes the server has access to representative, clean data, which is unrealistic in practical FL deployments, particularly under non-i.i.d. conditions. DeFL [13] detects malicious clients by analyzing the internal structure of the entire model through a federated gradient norm vector, capturing fine-grained differences in client updates. FLDetector [12] measures the consistency between each client's update and a predicted update generated by the server using historical trends. Similarly, FedDefender [11] employs differential testing to expose behavioral discrepancies among client models. While these methods show promise, they require unrestricted access to complete model updates from each client, which increases the risk of privacy leakage [14]. Moreover, their effectiveness degrades significantly in non-i.i.d. settings [17].

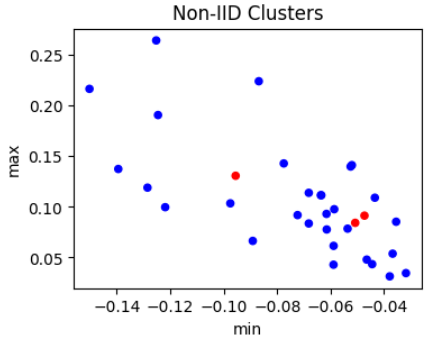
Certified robustness methods like weight smoothing [21], group ensemble [22], and adversarial training [16]—improve robustness against backdoor attacks but incur high computational costs, limiting their practicality in resource-constrained FL settings. Robust Learning Rate (RLR) [9] was proposed to defensively adjust the server's learning rate based on agents' model updates. While RLR effectively mitigates backdoor attacks, it inadvertently reduces the validation accuracy of the trained model, a drawback that becomes more pronounced in non-i.i.d settings. Furthermore, RLR fails to defend against backdoor attacks when a substantial number of FL agents are malicious.

Lockdown [17] is a pruning-based defense that mitigates the poison-coupling effect in FL through isolated subspace training. It assigns random subspaces to clients and employs subspace pruning and recovery to separate benign from malicious updates. A quorum consensus mechanism is then used to filter suspicious parameters from the global model. While Lockdown demonstrates strong defense performance and enhances communication efficiency, our study shows that its effectiveness degrades significantly for smaller models, where the consensus mechanism becomes less reliable.

In contrast, we present an efficient defense technique that reduces both computational overhead and privacy risks by inspecting a small fraction of model updates. Notably, when combined with RLR, our method outperforms state-of-the-art defenses across both i.i.d. and non-i.i.d. settings, even in the presence of a large number of malicious agents.



(a) i.i.d. setting



(b) non-i.i.d. setting

Fig. 1: Scatter plots of minimum and maximum bias updates from each FL agent during one training round. Red and blue indicate malicious and benign agents.

III. Efficient Detection and Mitigation of Backdoor Attacks

A. Model Update Analysis

We first analyzed the feasibility of detecting backdoor attacks from model updates in an FL setting. Using the Flower [23] library, we trained a CNN model across K agents, 10% of which were malicious. These agents poisoned 50% of base class samples with a trojan pattern and relabeled them as the target class. We adopted two experimental setups based on [9].

1) I.I.D Setting: We uniformly distributed the CIFAR10 dataset among 40 agents to train an AlexNet-style model [9]. We analyzed the parameter updates from both a malicious and a benign agent in each training round, which included 536,768 weights and 842 biases. Notably, the final 10 values, corresponding to the output layer biases, showed clear differences between the two agents. We found that the minimum and maximum values of these bias updates are effective features for identifying malicious agents, as illustrated in Figure 1a. This finding was validated over 50 repeated runs.

2) Non-I.I.D Setting: Next, we used the Federated EMNIST dataset from the LEAF benchmark [24], where digits 0–9 are distributed across 3,383 agents with non-i.i.d. data distributions. In each FL round, 10% of the agents were selected to train a LeNet-style model [9], each producing updates with 1,199,648 weights and 234 biases. Unlike the i.i.d. case, distinguishing between bias updates from malicious and benign agents proved more difficult. This challenge is reflected in the scatter plot of the minimum and maximum bias values (Figure 1b), which illustrates one of the most difficult cases observed across 50 runs. Despite this, Section V shows that our hybrid method, combining statistical outlier detection with RLR [9], performs robustly in non-i.i.d. settings.

B. Outlier Detection with Partial Model Inspection

We developed a Bias Outlier Detection (BOD) technique that uses statistical features to characterize each agent's bias updates in the final neural network layer. Outliers are detected using percentile-based thresholds computed across all agents in each FL round. An agent is classified as benign if its bias update's minimum and maximum fall within specified percentile bounds and either its standard deviation or skewness remains below the threshold; otherwise, it is flagged as malicious.

Algorithm 1 Bias Outlier Detection (BOD)

Require: Model updates $U \in \mathbb{R}^{d \times N}$ from N agents, percentiles p_1, p_2, p_3, p_4 for threshold calculation.
Ensure: Predicted benign set B, malicious set M

for $j = 1$ to N **do**
  Compute $x_j^{\min}, x_j^{\max}, x_j^{\text{std}}, x_j^{\text{skew}}$ from $U[:, j]$
end for
Compute thresholds $\theta_{\min}, \theta_{\max}, \theta_{\text{std}}, \theta_{\text{skew}}$ as the p_1th, p_2th, p_3th, and p_4th percentiles of the sets $\{x_j^{\min}\}$, $\{x_j^{\max}\}$, $\{x_j^{\text{std}}\}$, and $\{x_j^{\text{skew}}\}$, respectively.
Initialize benign set $B \leftarrow \emptyset$, malicious set $M \leftarrow \emptyset$
for $j = 1$ to N **do**
  if $x_j^{\min} > \theta_{\min}$ **and** $x_j^{\max} < \theta_{\max}$ **and** $(x_j^{\text{std}} < \theta_{\text{std}}$ **or** $x_j^{\text{skew}} < \theta_{\text{skew}})$ **then**
    $B \leftarrow B \cup \{j\}$
  else
    $M \leftarrow M \cup \{j\}$
  end if
end for
return (B, M)

In our outlier detection algorithm (Algorithm 1), $\mathtt{d}$ denotes the dimension of each client's update vector, specifically, the number of bias parameters in the final layer of the model. Let $\mathtt{N}$ represent the number of participating agents, each contributing a $\mathtt{d}$-dimensional update. We compute thresholds for four statistical features: minimum, maximum, standard deviation, and skewness across all updates. The 75th percentile is used as a

TABLE I: FL Experiment Hyperparameters

Data distribution	i.i.d.	i.i.d.	non-i.i.d.
Dataset	CIFAR10/ FMNIST	CIFAR10	Federated EMNIST
Models	AlexNet-style, ResNet-9, LeNet-style	AlexNet-style	LeNet-style
Attack type	Backdoor	DBA	Backdoor
Total agents	10	40	3,383
Agents selected	10	40	33
Local epochs	2	2	10
Batch size	256	256	64

TABLE II: Validation accuracy (Acc.) and attack Success Ratio (ASR) in percentage, evaluated under a 10% attack ratio with i.i.d. partitioned datasets.

Aggregation	**CIFAR10/ AlexNet-style**		**CIFAR10/ ResNet-9**		**FMNIST/ LeNet-style**	
	Acc.	ASR	Acc.	ASR	Acc.	ASR
FedAvg (baseline)	79.5	91.0	85.1	43.2	93.4	100.0
RLR	54.1	**3.6**	57.5	8.0	92.5	0.0
Lockdown	53.7	69.5	85.1	**6.3**	89.9	69.0
Mkrum	79.5	99.8	76.1	47.7	93.3	99.5
BOD	73.7	8.3	77.9	12.4	92.2	**0.7**
BOD-hybrid	74.2	7.3	78.2	8.3	91.9	1.8

lower bound for the minimum feature to exclude agents with abnormally large negative values, often indicative of malicious behavior. For the other features, 50th percentile thresholds capture the central tendency of benign updates. This percentile-based approach balances sensitivity and robustness, enabling dynamic detection of statistical outliers in heterogeneous update distributions. The method is computationally efficient, model-agnostic, and well-suited for real-world FL systems.

C. Backdoor Mitigation

In each FL training round, the server applies **BOD** to filter out potentially malicious updates before performing federated averaging. While this may exclude some benign updates due to false positives, our experiments (Section V) show that training remains robust and achieves high validation accuracy. To better handle non-i.i.d. data and reduce the impact of false negatives, we introduce a hybrid strategy (**BOD-hybrid**) that combines BOD with the **RLR** method [9]. After BOD filters suspect updates, RLR adjusts the server's learning rate based on the sign of each remaining update. For each parameter, the server sums the signs and compares the result to a threshold θ; if the sum is below θ, suggesting low consensus, the learning rate is inverted to steer training away from adversarial directions.

IV. Experiment Setup

Datasets and Models: We conducted our experiments using an FL setup implemented with Flower [23], a flexible, framework-agnostic platform for building FL systems. All models were built in PyTorch and trained for 200 FL rounds using SGD (local learning rate = 0.1, server learning rate = 1.0). We evaluated AlexNet-style CNN and ResNet-9 on the CIFAR-10 dataset, and used LeNet-style CNN for both FMNIST and Federated EMNIST. CIFAR-10 and FMNIST were i.i.d.-partitioned, while Federated EMNIST (from the LEAF benchmark [24]) followed a non-i.i.d. distribution. Additional experimental settings are summarized in Table I. All experiments were conducted on an NVIDIA DGX A100 system using a single NVIDIA A100 GPU (40 GB), AMD EPYC 7742 64-core CPUs, and 503 GB of system RAM. Our implementation is publicly available at https://github.com/cloudsyslab/federated-learning.

Attack Methods: As in prior studies [9], [17], each malicious agent poisons 50% of its local data. By default, 10% of the agents in the system are adversarial, and we also study the impact of increasing this attack ratio on defense performance. For the Federated EMNIST dataset, the backdoor attack causes the model to misclassify digit 1s as 7s (target class) by adding a plus sign to the top-left corner of images. For FMNIST, the attack maps sandals to sneakers using the same pattern. For CIFAR10, the attack causes the model to misclassify dogs as horses. We evaluate both standard Backdoor Attacks and Distributed Backdoor Attacks (DBA), where the plus pattern is split across 4 malicious agents.

Competing Defense: We compare BOD and BOD-hybrid against the baseline FedAvg and state-of-the-art defense methods, including Mkrum [7], RLR [9], and Lockdown [17]. For RLR, the learning threshold parameter θ is set to 4 for CIFAR10/FMNIST with 10 agents, 8 for CIFAR10 with 40 agents, and 7 for Federated EMNIST with 33 agents. These values offer the best balance between backdoor mitigation and validation accuracy [9]. Following the Lockdown [17] paper, we set the quorum consensus threshold to half the number of agents participating in each FL round.

Evaluation Metrics: These include (1) validation accuracy, which measures the percentage of clean samples correctly classified by the global model, and (2) attack success ratio (ASR), which measures the percentage of trojaned samples misclassified as the target label. Although our method may exclude some benign updates due to false positives, we focus primarily on validation accuracy and ASR, as these best reflect the real-world effectiveness of the defense.

V. Results

A. With I.I.D Setting

The results presented in Table II highlight the effectiveness of our BOD and BOD-hybrid methods in

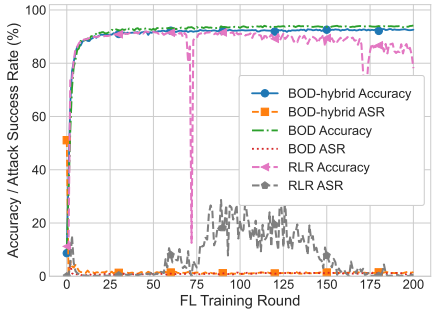
mitigating backdoor attacks under i.i.d. settings. Across all models and datasets, our methods consistently achieve low attack success ratios (ASR) while maintaining high validation accuracy. FedAvg performs poorly in all cases failing to defend against backdoor attacks, with ASRs greater than 90%. The performance of RLR varies notably across datasets. On FMNIST, it achieves an ASR of 0% while maintaining high validation accuracy. However, on CIFAR10, despite achieving low ASRs of 3.6% (AlexNet) and 8.0% (ResNet-9), it significantly degrades the validation accuracy, dropping it to as low as 54.1%. In contrast, our BOD-hybrid method delivers comparable ASRs of 7.3% and 8.3%, while preserving substantially higher validation accuracy at 74.2% and 78.2% for the AlexNet and ResNet-9 models, respectively. Mkrum fails across all scenarios, with ASRs exceeding 99%, indicating an inability to defend against backdoor attacks. Lockdown shows strong performance with larger models such as ResNet-9 but is ineffective with smaller models, yielding high ASRs. Both BOD and BOD-hybrid maintain a strong balance between robustness and utility across all model configurations.

B. With non-I.I.D Setting and Varying Attack Ratios

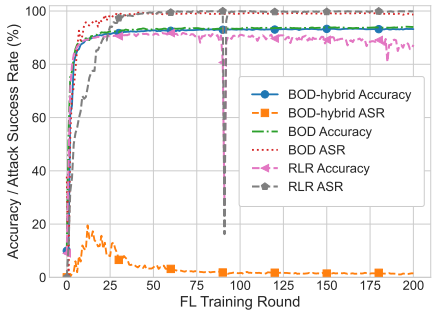
We evaluated our defense methods under non-i.i.d. data distributions using the Federated EMNIST dataset and the LeNet-style CNN model. Figures 2a and 2b show validation accuracy and attack success ratio (ASR) over training rounds for attack ratios of 10% and 30%. At the end of 200 rounds of FL training, RLR achieves a low ASR of 0.16% at 10% attack ratio, while sacrificing validation accuracy, which drops to 87.27%. RLR performs much worse at 30% attack ratio, with ASR rising to 99.89%. In contrast, BOD-hybrid shows consistent performance across both settings, maintaining low ASR (1.6% at 10% attack ratio, 1.5% at 30% attack ratio) and high validation accuracy (92.6% and 93.2%, respectively), demonstrating robustness to increased adversarial presence. While BOD performs best at 10% attack ratio (94.2% accuracy, 0.84% ASR), ASR rises to 98.7% at 30% attack ratio, indicating vulnerability under higher attack pressure. Overall, BOD-hybrid provides the most reliable defense in non-i.i.d. settings.

C. Distributed Backdoor Attack

Table III shows the performance of various defense methods under a distributed backdoor attack using the CIFAR10 dataset and the AlexNet-style CNN model. The baseline FedAvg method fails to defend against the attack, with a high ASR of 66.0%. RLR achieves an ASR of 8.5% but at the cost of reduced accuracy (71.8%). Mkrum maintains relatively high accuracy (77.0%) and a reasonable ASR of 4.9%. Lockdown performs worst among all methods, with a severe drop in validation accuracy to 13.6% and an ASR of 84.2%, indicating that it fails entirely in this setting. Our BOD method provides a better balance, reducing ASR to 4.5% while preserving 72.8% accuracy. Notably, BOD-hybrid achieves the best overall defense, lowering ASR to 3.8% and maintaining 72.6% accuracy, demonstrating strong resilience against sophisticated distributed attacks.



(a) 10% attack ratio.



(b) 30% attack ratio.

Fig. 2: Validation Accuracy and Attack Success Ratio (ASR) over FL training rounds in a non-i.i.d. setting (Federated EMNIST) under varying attack ratios.

TABLE III: Validation Accuracy and Attack Success Ratio (ASR) under a distributed backdoor attack.

Aggregation	Validation Accuracy (%)	ASR (%)
FedAvg (baseline)	78.7	66.0
RLR	71.8	8.5
Lockdown	13.6	84.2
Mkrum	77.0	4.9
BOD	72.8	4.5
BOD-hybrid	72.6	**3.8**

D. Overhead Analysis

Figure 3 compares the average time per FL training round using various defense methods on AlexNet-style and ResNet-9 models with the CIFAR10 dataset. **FedAvg** incurs the least overhead as it performs simple aggregation with no defense. Mkrum requires $O(N^2d)$ pairwise distance computations for N agents (d is the total model dimension). Lockdown's subspace training, pruning, and consensus checks make the method most costly. In

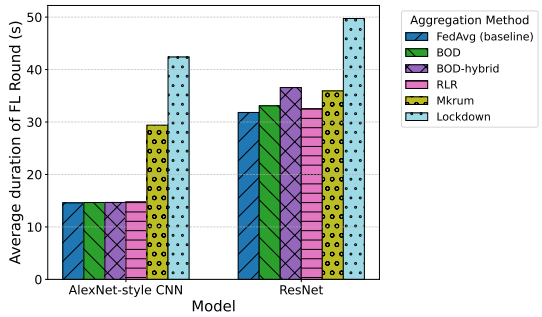


Fig. 3: Impact of defense methods on training efficiency.

contrast, **RLR** and our methods are significantly more efficient. RLR's sign aggregation runs in $O(Nd)$ time, while **BOD** only inspects the $b \ll d$ bias parameters in the final layer. It computes client-level statistics in $O(Nb)$ and sorts them in $O(N \log N)$, yielding a total complexity of $O(Nb + N \log N)$, with Nb dominating in practice. The **BOD-hybrid** approach introduces only a minor additional cost of $O(N'd)$, where $N' < N$ since RLR is applied only to the subset of agents, which are not filtered out by BOD. Overall, BOD and BOD-hybrid provide greater efficiency while preserving robustness, making them well-suited for practical FL deployments.

VI. Conclusion

We developed a lightweight and effective defense against backdoor attacks in federated learning based on partial inspection of model updates. By analyzing simple statistics of final-layer bias parameters, our method efficiently identifies malicious agents with minimal overhead. We also introduced a hybrid strategy, which synergistically integrates a robust learning rate method to enhance robustness in non-i.i.d. and high-attack rate scenarios. While we do not provide formal guarantees, empirical results demonstrate strong robustness across datasets and attack intensities. Importantly, our methods are computationally efficient and practical for real-world deployments. Future work will explore dynamic defense adaptation and broader attack classes.

Acknowledgment

The research was sponsored by the Army Research Office and accomplished under Grant Number W911NF-23-1-0007.

References

[1] K. Doshi and Y. Yilmaz, "Federated learning-based driver activity recognition for edge devices," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.

[2] N. A. Tu, A. Abu, N. Aikyn, N. Makhanov, M.-H. Lee, K. Le-Huy, and K.-S. Wong, "Fedfslar: A federated learning framework for few-shot action recognition," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

[3] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. of the 36th International Conference on Machine Learning*, 2019.

[4] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*. PMLR, 2020.

[5] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

[6] 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," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[7] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017.

[8] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018.

[9] M. S. Ozdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *Proc. of the AAAI conference.*, 2021.

[10] K.-H. Chow, L. Liu, W. Wei, F. Ilhan, and Y. Wu, "Stdlens: Model hijacking-resilient federated learning for object detection," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[11] W. Gill, A. Anwar, and M. A. Gulzar, "Feddefender: Backdoor attack defense in federated learning," *arXiv preprint arXiv:2307.08672*, 2023.

[12] Z. Zhang, X. Cao, J. Jia, and N. Z. Gong, "Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients," in *Proc. of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[13] G. Yan, H. Wang, X. Yuan, and J. Li, "Defl: Defending against model poisoning attacks in federated learning via critical learning periods awareness," in *Proc. of the AAAI Conference*, 2023.

[14] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, 2019.

[15] M. Alfarra, J. C. Pérez, E. Shulgin, P. Richtárik, and B. Ghanem, "Certified robustness in federated learning," *arXiv preprint arXiv:2206.02535*, 2022.

[16] D. Shah, P. Dube, S. Chakraborty, and A. Verma, "Adversarial training in communication constrained federated learning," *arXiv preprint arXiv:2103.01319*, 2021.

[17] T. Huang, S. Hu, K.-H. Chow, F. Ilhan, S. Tekin, and L. Liu, "Lockdown: backdoor defense for federated learning with isolated subspace training," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10 876–10 896, 2023.

[18] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Annual Network and Distributed System Security Symposium*, 2018.

[19] 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)*, 2020.

[20] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020.

[21] C. Xie, M. Chen, P.-Y. Chen, and B. Li, "Crfl: Certifiably robust federated learning against backdoor attacks," in *International Conference on Machine Learning*, 2021.

[22] X. Cao, Z. Zhang, J. Jia, and N. Z. Gong, "Flcert: Provably secure federated learning against poisoning attacks," *IEEE Transactions on Information Forensics and Security*, vol. 17, 2022.

[23] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, H. L. Kwing, T. Parcollet, P. P. d. Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[24] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.