

On the Effectiveness of Invisible Backdoor Attacks in Federated Learning

Citation for published version (APA):

Marinus, M., Tsouvalas, V., & Meratnia, N. (2025). On the Effectiveness of Invisible Backdoor Attacks in Federated Learning. In G. Wang, Z. Yan, K.-C. Li, & Y. Wu (Eds.), *Ubiquitous Security: 4th International Conference, UbiSec 2024, Changsha, China, December 29–31, 2024, Revised Selected Papers* (pp. 174-190). (Communications in Computer and Information Science (CCIS); Vol. 2469). Springer.
https://doi.org/10.1007/978-981-96-4836-8_13

Document license:
CC BY-ND

DOI:
[10.1007/978-981-96-4836-8_13](https://doi.org/10.1007/978-981-96-4836-8_13)

Document status and date:
Published: 16/05/2025

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

On the Effectiveness of Invisible Backdoor Attacks in Federated Learning

Mattjis Marinus, Vasileios Tsouvalas, and Nirvana Meratnia

Eindhoven University of Technology, The Netherlands

m.j.marinus@student.tue.nl, {v.tsouvalas,n.meratnia}@tue.nl

Abstract. Federated Learning (FL) enables collaborative training of machine learning models across multiple devices, while preserving data privacy. However, it also introduces vulnerabilities to backdoor attacks, where malicious updates can corrupt the global model. This work focuses on the largely unexplored domain of Federated invisible Backdoor Attacks (FiBA s), which use visually indistinguishable triggers to manipulate model behavior without being detected. We investigate the feasibility and effectiveness of these attacks, considering the unique challenges posed by FL, such as limited local training time and the need for model update aggregation. Our study presents a comprehensive evaluation of the attack success rate (ASR) of invisible BAs under various settings and defenses. Based on our observations, we propose a backdoor trigger hiding technique based on low model attention regions to improve attack resilience in federated settings. Our findings provide critical insights into the optimization of invisible BAs in FL and highlights the need for robust defense mechanisms to safeguard FL systems.

Keywords: federated learning; backdoor attacks; invisible backdoor triggers; distributed triggers; FiBA

1 Introduction

Federated Learning (FL) is a distributed machine learning paradigm that facilitates collaborative training of deep neural networks across multiple nodes (clients) without the need to share raw data [19]. In FL, model updates are derived on-device and communicated to a central server, which aggregates these updates to create a unified global model. This approach is particularly beneficial for applications requiring high data privacy, such as finance and healthcare, as it improves model performance without the need to share sensitive data. While FL offers significant advantages in terms of data privacy and collaborative learning, it also introduces unique security vulnerabilities, among which backdoor attacks that are able to manipulate the global model by embedding malicious triggers in local updates [18, 22, 25]. Exploiting the distributed nature of FL, these attacks manipulate the model’s behavior when a specific triggers is present on its input, posing significant risks in critical sectors where data integrity is paramount.

While Backdoor Attacks (BAs) in FL have been widely studied, invisible backdoor triggers, which are visually indistinguishable from normal data, have

mainly received attention in centralized settings. However, in FL, an attacker can briefly access the data, inject invisible triggers into the locally stored data, and then leave the system, leading to catastrophic consequences after training. This is particularly concerning in cross-silo settings, where clients often represent large organizations such as banks [27]. Despite its importance, the effectiveness and challenges of deploying invisible backdoor triggers in FL remain largely unexplored. Existing research has primarily focused on simpler, more obvious pixel patterns, neglecting the complexities associated with invisible triggers [2, 11]. Moreover, the limited local training time and need to compete with honest clients’ updates during model aggregation in FL, exacerbate the difficulty of successful embedding and maintaining these subtle triggers.

This work aims to bridge this gap by investigating performance of invisible BAs in FL. Specifically, we assess how the visual stealthiness of backdoor triggers impacts their learnability and effectiveness, given the unique constraints of FL training. Concisely, our contributions are as follows:

- We explore the effectiveness of various invisible BA triggers under the inherent limitations of FL, such as limited local training time and server-side model update aggregation.
- We establish a benchmark of invisible BAs across various federated settings and backdoor defence mechanisms, emphasizing the significance of *Federated Invisible Backdoor Attacks* (FIBAs).
- We explored advanced stealthiness boosting techniques in FIBAs, providing new insights into optimizing BAs in FL. From our observations, we propose a novel trigger-hiding technique to improve FIBAs attack success rate.

2 Related Work

Invisible Backdoor Attacks. Backdoor attacks have evolved significantly since the initial demonstrations such as BadNet [12], which inserted visible patterns into training data. These early attacks were effective but easily detectable. Recent advancements have focused on enhancing the stealthiness of these attacks, making the triggers less visible to human inspection, while maintaining high attack success rates. Invisible BAs can be broadly classified into blend-based, transformation-based, and optimization-based categories. Blend-based attacks, like Blended [6], merge predetermined triggers with the original images, making them stealthier than simpler pixel-based triggers, though still somewhat visible. Transformation-based attacks employ sophisticated algorithms to subtly modify the data. Examples include the LSB attack [16], which modifies the least-significant bits of an image, and WaNet [21], which uses warping fields to create slight distortions. These methods improve stealth by creating complex patterns that are harder to reverse engineer and detect. Optimization-based attacks generate triggers through a series of optimization steps, often using deep neural networks. Techniques such as LIRA [8] embed information within images using auto-encoders or solve optimization problems to maximize normal data performance while embedding a backdoor, minimizing perceptual differences and making triggers virtually undetectable to the human eye. Apart from

BAs, backdoor triggers can also be categorized into semantic and artificial triggers, with semantic triggers leveraging naturally occurring features within the data to mislead the model, while artificial ones involve adding specific patterns to the data.

Federated Backdoor Attacks. The distributed nature of FL training allows attackers to introduce backdoors through local updates. Studies [11] have shown that distributed artificial triggers can achieve high success rates by exploiting the variation in updates across multiple malicious clients. However, research on backdoor attacks in FL has primarily focused on obvious, pixel-based triggers, leaving the potential of visually stealthy triggers underexplored. Existing work, such as Federated Invisible Backdoor Attacks (FedFIBA) [32], has highlighted the challenges in maintaining the stealthiness of triggers while ensuring their effectiveness in FL. FedFIBA explored transformation-based invisible triggers and used optimization techniques to balance stealthiness and effectiveness, incorporating visual perception scores to adjust trigger visibility. Our work differentiates itself by specifically focusing on developing and evaluating backdoor triggers that are both effective and visually undetectable in the FL context. By addressing gaps in the current research, we aim to enhance the understanding and robustness of BAs mechanisms in FL.

Defense Mechanisms against BAs. As interest in backdoor attacks in FL has grown, so has the development of defense mechanisms in FL. These mechanisms can be broadly categorized into model updates comparison, robust model aggregation, and model inspection techniques. Model updates comparison methods, such as FoolsGold [9] and CONTRA [1] detect malicious updates by comparing the cosine similarity of updates from different clients, aiming to identify colluding attackers. Clustering techniques, such as PCA [30] are proposed to determine groups of malicious updates. Multi-KRUM [5] uses distances between model updates to determine the most representative update. Robust model aggregation techniques, like Median and Trimmed Mean, focus on reducing the impact of anomalous updates by calculating the median or trimming extreme values, respectively, to create a more resilient global model. Model inspection methods, including Neural Cleanse [31] and GradCam [26], analyze the model during or after training to identify potential backdoors by visualizing neuron activations or activation maps. Lockdown [14], inspired by model pruning, limits the influence of each client’s updates using subspace masks and consensus fusion. Despite these advancements, defenses against invisible backdoor attacks in FL are still nascent. Most existing defenses target obvious attacks and may not effectively address visually stealthy backdoors. Our work pioneers the evaluation of invisible backdoor attacks against a broad spectrum of FL defenses, providing essential insights and proposing new strategies for enhancing FL security.

3 Methodology

Problem Formulation. While invisible backdoor triggers have proven effective in non-distributed ML, their applicability in FL faces two unique challenges: (i) the need for small, invisible triggers makes them harder for models to learn, and

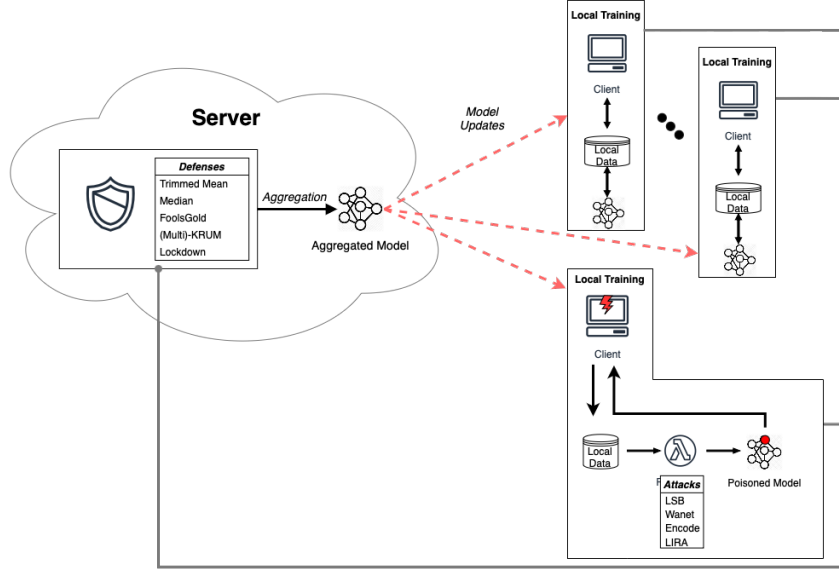


Fig. 1: FL training stage in presence of FIBAs. Attackers inject their backdoor trigger during local training stage, while BA mitigation is performed during the server-side aggregation process.

(ii) during FL training, attackers’ model updates compete with those of honest clients, limiting the global model’s ability to learn the trigger. To highlight these challenges, we examine *artificial* backdoor attacks, which embed backdoor triggers into the global model by steering it to classify samples with artificial backdoor triggers as a target class, t . Specifically, attackers inject the trigger into locally-stored data, either through a transformation function or via a specialized model that generates a new image, creating the “*poisoned*” dataset ($\mathcal{D}_p$). Subsequently, during the local training phase, attackers’ local training is altered to optimize the loss on both the original “*clean*” dataset ($\mathcal{D}_c$) and the “*poisoned*” dataset ($\mathcal{D}_p$). Formally, the attacker’s optimization problem is defined as:

$$\min \mathcal{L}(\mathcal{D}_c, \mathcal{D}_p, h_\theta) = \sum_{i=1}^{|\mathcal{D}_c|} \mathcal{L}_{CE}(h_\theta(x_i), y_i) + \sum_{j=1}^{|\mathcal{D}_p|} \mathcal{L}_{CE}(h_\theta^*(x_j), t) \quad , \quad (1)$$

where $\mathcal{L}$ is the supervised loss term over attacker “*clean*” ($\mathcal{D}_c$) and “*poisoned*” ($\mathcal{D}_p$) datasets, given a model h with parameter θ . With θ^* we refer to the model parameter once training on $\mathcal{D}_c$ is complete in a given federated round.

3.1 FiBAs: Federated Invisible Backdoor Attacks

Training Overview. The training pipeline starts with the server randomly initializing a neural network, h_θ , which is sent to all clients. Non-attacker clients perform local training by executing several local SGD steps on their data. Attackers, however, modify a portion of their local dataset with an invisible backdoor trigger and use this poisoned data during local SGD steps. In our work, we

assume attackers can temporarily or continuously access a client’s data, inject invisible triggers, and leave the device intact to continue training. The server then aggregates the model updates from all clients. If no defense is applied, FedAvg [19] serves as the default aggregation strategy. Alternatively, a defense mechanism may be used during aggregation to mitigate backdoor attacks by filtering or limiting specific client updates. An overview of the FL training stage with BAs is shown in Figure 1.

Poisoning Strategy. An attacker can choose to either poison all classes to a single target class (t) (*all-to-one*) or poison a specific source class to the target class (t) (*one-to-one*). With both poisoning strategies being valid, the ultimate choice depends on the attacker’s goals. While *one-to-one* attacks are stealthier — introducing focused triggers and thus affecting fewer model parameters — our preliminary experiments showed that *one-to-one* BAs exhibit high variance and volatility. In FiBA, we adopt an *all-to-one* poisoning strategy, similar to [8, 16, 21, 32], to evaluate the effectiveness of invisible BAs. Apart from the choice of poisoning target class, the attacker’s poisoning rate (p) is a crucial aspect of all BAs, which determines the proportion of locally stored data that the attacker will be poisoning. We set p to range between 0.1 and 0.5, following the state-of-the-art [2, 8, 21, 24].

3.1.1 Adapting Invisible BAs for FL

We investigate state-of-the-art invisible BAs in FL, namely LSB, Encoder, WaNet, and LIRA attacks, against common FL defense mechanisms, to provide insights into the challenges and defenses for sophisticated backdoor threats in FL. As the aforementioned invisible BAs are not specifically tailored for FL, we introduce the modifications and assumptions under which we can perform these attacks in the federated settings:

- **LSB Attack [16]** alters the least significant bits of image pixels to encode a string, making it visually undetectable. In FL, multiple attackers must agree on the encoded string beforehand to ensure each attacker creates the same local trigger.
- **WaNet Attack [21]** warps input samples using elastic image warping to create subtle distortions. For FL, attackers must agree on the hyper-parameters k and s and use the same sampling field s , drawn from a uniform distribution between -1 and 1 , to maintain consistent warping patterns.
- **Encoder Attack** uses an auto-encoder trained to hide a trigger. In contrast to [3, 20, 28, 29], where a colored image was utilized as a trigger, we opt to utilize a single bit-string, similar to [17], increasing the stealthiness of the BA. To assess the attack’s effectiveness in FL, we maintain the same auto-encoder architecture and pre-training process as [3, 20, 28, 29]. In FL, attackers either agree on a common noise pattern and train their auto-encoders separately or share a single auto-encoder model.
- **LIRA Attack [8]** uses adversarial training between a classification model and a transformation function T that applies the backdoor to images. The transformation function $T_\xi(x) = x + g_\xi(x)$ generates noise $g_\xi(x)$, with the ϵ

parameter controlling the stealthiness by limiting the l_∞ norm of the noise. In FL, attackers must coordinate to train the LIRA model on shared data before the FL training stage.

3.1.2 ASR Boosting Techniques in FiBAs

We further adapt widely-used ASR boosting techniques, typically applied to simple pattern triggers, within FiBA to investigate whether they achieve comparable results in FL.

- **Model Replacement.** *Model replacement* [2] boosts ASR by scaling an attacker’s model updates to overpower others during (weighted) averaging. This requires knowing the total number of training samples (n) and clients (m). At round t , the attacker’s update X is scaled to replace the global model g^t with w_j^{t+1} using $w_j^{t+1} \simeq \frac{n}{r_j} X - (m-1)g^t$, where r_j is the attacker’s sample count. By scaling by $\frac{n}{r_j}$ and subtracting $(m-1) \times g^t$, the attacker can dominate the next aggregation. However, can be easily inferred through channel listening or insider knowledge, the assumption that n can be guessed or inferred, is unrealistic [2].
- **Distributed Triggers.** Splitting the trigger into smaller parts and distributing them among attackers has been shown to increase the ASR of simple-pattern BAs [11,12]. Applying this technique to invisible BAs is more challenging since the triggers are either image-dependent or spread loosely across the image. To achieve a similar effect in FiBA, we divide images into equal-sized rectangles based on the number of attackers, padding any remaining space to the last rectangle if needed. Attackers poison their data samples as usual, then replace the parts of the image outside their assigned rectangle with the original counterparts, giving each attacker a distinct part of the trigger to learn.

3.2 Boosting FiBAs Stealthiness via Low Activation Triggers

Based on the fact that global model attention is distributed across various regions and features, we propose a strategy to increase the ASR of FiBAs by hiding the trigger in low-activation areas of the global model. This approach aims to minimize competition during server-side model aggregation, as high-attention regions face greater competition from honest clients’ updates. Our motivation stems from observing that invisible BAs like WaNet and LIRA, which often overlap with identifying features, perform worse than simpler attacks like BadNet [12] in FL, as shown in Section 5.1.

Finding Low Activation Areas. To identify low activation areas, we utilize GradCAM [26] to extract regions with high activations from the global model received by the attacker each round. It is crucial to perform this extraction once the clients’ models have sufficiently converged, ensuring accurate information about important regions. We assess the model’s performance using embeddings from the penultimate layer, using the rank of embeddings as

Algorithm 1 Activation Mask Extraction Algorithm.

Require: model h , dataset $\mathcal{D}$, inputs' shape z

- 1: Initialize CAM: $\text{cam} \leftarrow \text{SmoothGradCAM++}(h, z, h.\text{layer1})$
- 2: Initialize Mask: $\text{activation_mask} \leftarrow \mathbf{0}_{16 \times 16}$
- 3: **for** each batch (x, y) in ds **do**
- 4: **for** each image x_i in batch (x, y) **do**
- 5: $\text{logits} \leftarrow \text{model}(x_i)$
- 6: $A_i \leftarrow \text{cam}(\arg \max(\text{logits}), \text{logits})$
- 7: $\text{activation_mask} += A_i$
- 8: **end for**
- 9: **end for**
- 10: Sort and create top half mask:
- 11: $\text{sorted_indices} \leftarrow \text{argsort}(\text{flatten}(\text{activation_mask}))$
- 12: $\text{top_half_mask} \leftarrow \mathbf{0}$
- 13: $\text{top_half_mask}[\text{sorted_indices}[0 : \frac{N}{2}]] \leftarrow 1$

a proxy for generalization quality [10]. Formally, we extract the embeddings $\mathcal{Z}$ from $\mathcal{D}$ using the global model h_θ , and afterwards compute the score, $\mathcal{E}$ using $\exp(-\sum_{j=1}^{m_{\mathcal{Z}}} r_j \log r_j)$, where r_j denoting the ranking of j -th singular value of $\mathcal{Z}$ ($\frac{\sigma_j}{|\sigma_{\mathcal{Z}}|_1}$) and $m_{\mathcal{Z}}$ denotes the minimum dimension of embeddings.

Once the RankMe score reaches a certain threshold (τ), we construct an activation mask by processing locally stored samples through the model and using SmoothGradCam++ [23] to extract the activation maps. These activation maps are then summed to form a comprehensive activation mask. The mask is sorted, and the indices of the top half of the most important regions are collected. These regions, identified as the top half, are marked with 1s, while the remaining regions are marked with 0s. This binary mask, denoted as *low activation mask*, now highlights the regions that receive the least attention from the global model that can be exploited by attackers to hide their triggers.

Low activation trigger in FiBAs. For the LSB attack, a *low activation trigger* can be constructed by simply skipping bit-flipping in high activation regions marked by the mask. Alternatively, in the encoder-based attack, the mask can be used to penalize differences between the original and poisoned image in high attention areas more heavily, thus forcing the triggers into low activation regions. Integration of low activation masks into WaNet and LIRA attacks is left for future work.

4 Experimental Setup

Datasets & Models. We focus on vision classification tasks and use publicly available image classification datasets, namely CIFAR-10 and CIFAR-100 [15], in which we apply standard data augmentation policies [7]. For our data splitting procedure, we partitioned the data across clients in a non-overlapping fashion, utilizing a Dirichlet distribution [14] over classes, denoted as $\text{Dir}(a)$, where a refers to the distribution concentration. To simulate non-IID settings, we fix

$\alpha=0.5$, essentially controlling the class distribution per client (referred to as σ). We fix any randomness in the aforementioned partitioning process via a seed value, ensuring identical data distributions across our experiments to enable direct comparison. Across our experiments, we utilized ResNet-18 [13], training the model from scratch (no pre-trained weights), where each client completed 5 local training epoch per round ($E=5$) with a batch size of 64, using the Adam optimizer and learning rate of $1e-3$.

FL Settings & Parameters. We used Flower [4] and controlled the federated setting with the following parameters: number of clients - N , percentage of attackers - m , number of rounds - R , local train epochs - E , clients’ participation rate per round - ρ , and client class concentration - σ . Furthermore, we introduced BA-specific parameters, including the poison rate (p) between non-target and target (t) classes, which we set to 0.5. Since $E = 5$ limits the time attackers have to learn complex patterns, a low p could be ineffective, while with (temporary) data access, the extent of poisoning is less critical as the backdoor triggers remain visually imperceptible. For BadNet we placed 5 white pixels in the lower right corner, while for LSB we use a trigger’s string size (LSB_n) of 700. Since the image size is 32×32 , an LSB_n of 700 covers the image without reducing its visual stealthiness (all pixels altered at most twice). For WaNet, we set the warping’s control-grid size (WaNet_k) to 6 and its magnitude (WaNet_s) to 0.75. For all Encoder-based attack, an auto-encoder was trained for 100 rounds prior to the FL training stage with Enc_n of 200. Lastly, in LIRA, we set the noise magnitude (LIRA_ϵ) to 0.1 — enhancing trigger’s dominance in images — as there are no visually perceptible differences in LIRA_ϵ values from 0.1 to $1e^{-3}$ [8].

Baselines. We analyzed effectiveness of all FiBAs by measuring final model accuracy (Acc) — the percentage of correctly classified samples on the test set—and attack success rate (ASR) — the percentage of poisoned samples (excluding the target class) classified as the target class on the test set. Additionally, we evaluated FiBAs performance against prevalent defense mechanisms in FL for BAs, such as Median, Trimmed Mean for robust aggregation, and FoolsGold, PCA, KRUM, and Multi-KRUM from behavior detection approaches, while we included BadNet [12] to show the differences simple trigger patterns. Across our experiments, we used the default parameters for all defence mechanism. Lastly, we incorporate a visual perception score, specifically LPIPS [33], to assess the visual stealthiness of the attacks (lower is better). We report the mean Acc and ASR across our experiments.

5 Results

5.1 FiBAs effectiveness in FL

In this section we analyze *how invisible backdoor attacks, initially designed for centralized ML, perform in federated setting and what is the performance difference between BAs with invisible backdoor triggers as opposed to simple BAs with visible trigger patterns*. For this, we performed experiments for all considered FiBAs with 10 clients under non-i.i.d. settings in CIFAR-10 with a varying number of attackers. We present our findings in Figure 2 and Table 1.

Table 1: Performance evaluation of FIBAs on CIFAR-10. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

Method	$m = 0.1$		$m = 0.2$		LPIPS($\downarrow$) [33]	Anomaly Index($\downarrow$) [31]
	Acc	ASR	Acc	ASR		
FedAvg	84.20 $\pm$ 0.27	-	84.20 $\pm$ 0.27	-	-	-
BadNet	84.00 $\pm$ 0.31	94.96 $\pm$ 0.81	83.53 $\pm$ 0.24	97.52 $\pm$ 0.40	1.3×10^{-3}	8.80
LSB	83.37 $\pm$ 0.33	41.59 $\pm$ 6.19	83.60 $\pm$ 0.28	65.43 $\pm$ 3.93	7.83×10^{-5}	1.61
WaNet	83.33 $\pm$ 0.31	44.41 $\pm$ 3.50	83.75 $\pm$ 0.15	82.05 $\pm$ 3.59	6.4×10^{-3}	3.12
Encoder	83.44 $\pm$ 0.27	29.44 $\pm$ 5.90	83.79 $\pm$ 0.20	88.59 $\pm$ 4.01	3.13×10^{-4}	1.10
LIRA	83.71 $\pm$ 0.41	10.52 $\pm$ 1.22	83.93 $\pm$ 0.27	41.57 $\pm$ 4.66	1.06×10^{-4}	0.67

The LSB, WaNet, and Encoder attacks perform similarly, with their ASR gradually increasing over subsequent rounds, whereas LIRA performs the worst with minimal ASR improvement. Compared to the BadNet attack, there is a significant difference in ASR; however, the invisible BAs have not yet converged and might improve with more training rounds, indicating they suffer more in the aggregation process. Local attacker models for BadNet, WaNet, LSB, and Encoder all reach approximately 0.99 ASR, suggesting that performance differences arise mainly from server-side aggregation. None of the attacks significantly impact the model’s accuracy, maintaining values close to FedAvg without attacks, which is crucial for the feasibility of BAs. Increasing m to 20% results in a sharp ASR increase for all attacks, with WaNet and Encoder reaching over 80% ASR and LIRA improving by approximately 30%, while Acc remains unaffected. This indicates that invisible BAs are feasible in FL settings but require more attackers to achieve high ASRs, when compared to centralized settings. Lastly, from both LPIPS and Anomaly Index scores depicted in Table 1, we notice that values are significantly smaller than BadNet, indicating that such these attacks are stealthier than simple backdoor trigger patterns.

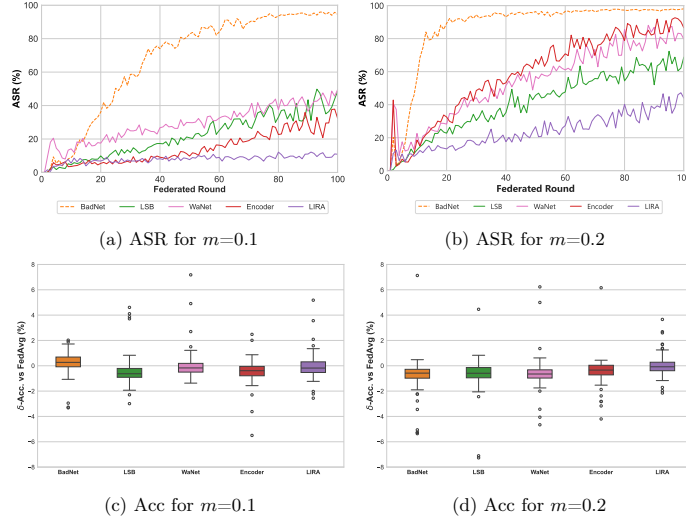


Fig. 2: FIBAs on CIFAR-10 at different attacker rates (m). Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

Table 2: Evaluation of FIBAs against Defense Mechanisms for BAs on (a) CIFAR-10 and (b) CIFAR-100. Parameters are $N = 10$, $R = 100$, $E = 5$, $\sigma = 0.5$, and $t = 5$.

Attack	Acc						ASR					
	FedAvg	Median	Trimmed Mean	FoolsGold	KRUM	Multi-KRUM	FedAvg	Median	Trimmed Mean	FoolsGold	KRUM	Multi-KRUM
BadNet	84.00 $\pm$ 0.31	84.00 $\pm$ 0.31	83.44 $\pm$ 0.53	84.32 $\pm$ 0.17	65.49 $\pm$ 4.46	66.32 $\pm$ 5.76	94.96 $\pm$ 0.81	94.96 $\pm$ 0.81	94.16 $\pm$ 1.17	89.98 $\pm$ 1.87	15.29 $\pm$ 7.50	18.04 $\pm$ 12.37
LSB	83.37 $\pm$ 0.33	83.16 $\pm$ 1.68	84.52 $\pm$ 0.40	83.26 $\pm$ 0.35	53.81 $\pm$ 10.20	69.67 $\pm$ 1.89	41.59 $\pm$ 6.19	29.74 $\pm$ 12.35	60.03 $\pm$ 7.17	27.01 $\pm$ 3.08	23.84 $\pm$ 26.93	1.47 $\pm$ 0.45
WaNet	83.33 $\pm$ 0.31	83.30 $\pm$ 0.55	84.00 $\pm$ 0.61	84.58 $\pm$ 0.23	60.45 $\pm$ 7.84	68.24 $\pm$ 1.31	44.41 $\pm$ 3.50	35.69 $\pm$ 3.68	36.57 $\pm$ 5.99	31.63 $\pm$ 1.61	51.05 $\pm$ 30.42	12.08 $\pm$ 1.00
Encoder	83.44 $\pm$ 0.27	83.15 $\pm$ 0.89	84.36 $\pm$ 0.35	85.02 $\pm$ 0.18	65.19 $\pm$ 8.74	70.91 $\pm$ 1.17	29.44 $\pm$ 5.90	43.79 $\pm$ 4.61	35.25 $\pm$ 4.11	36.74 $\pm$ 5.07	45.28 $\pm$ 23.57	7.45 $\pm$ 1.12

(a) CIFAR-10 with $m = 0.1$

Attack	Acc						ASR					
	FedAvg	Median	Trimmed Mean	FoolsGold	KRUM	Multi-KRUM	FedAvg	Median	Trimmed Mean	FoolsGold	KRUM	Multi-KRUM
BadNet	56.74 $\pm$ 0.24	55.17 $\pm$ 0.18	55.39 $\pm$ 0.26	55.96 $\pm$ 0.21	35.66 $\pm$ 1.00	49.16 $\pm$ 0.54	92.07 $\pm$ 0.25	92.90 $\pm$ 0.23	93.00 $\pm$ 0.22	92.04 $\pm$ 0.29	0.19 $\pm$ 0.23	0.32 $\pm$ 0.13
LSB	56.39 $\pm$ 0.19	54.63 $\pm$ 0.69	55.58 $\pm$ 0.40	56.75 $\pm$ 0.18	35.76 $\pm$ 0.98	48.30 $\pm$ 0.52	54.57 $\pm$ 4.03	57.88 $\pm$ 22.18	59.24 $\pm$ 9.85	33.84 $\pm$ 3.97	0.54 $\pm$ 0.48	0.32 $\pm$ 0.08
WaNet	55.32 $\pm$ 0.23	55.23 $\pm$ 0.47	54.90 $\pm$ 0.27	56.70 $\pm$ 0.19	33.88 $\pm$ 1.41	48.62 $\pm$ 0.73	37.44 $\pm$ 2.57	69.58 $\pm$ 11.80	70.45 $\pm$ 7.71	44.75 $\pm$ 2.47	0.06 $\pm$ 0.05	0.20 $\pm$ 0.15
Encoder	56.18 $\pm$ 0.18	54.20 $\pm$ 0.35	55.66 $\pm$ 0.22	55.77 $\pm$ 0.19	35.77 $\pm$ 1.80	48.20 $\pm$ 0.51	55.88 $\pm$ 2.86	79.33 $\pm$ 7.87	90.13 $\pm$ 2.47	70.06 $\pm$ 2.03	0.13 $\pm$ 0.05	0.31 $\pm$ 0.18

(b) CIFAR-100 with $m = 0.2$

5.2 FiBAs Performance vs. Backdoor Defences

Here, we aim to investigate *to what degree can widely-used backdoor defenses in FL mitigate invisible backdoor attacks*. For this, we perform experiments, where we evaluate FiBAs against mechanisms from both major defence categories, namely robust aggregation (Median and Trimmed Mean) and behavior detection (FoolsGold, PCA, KRUM, and Multi-KRUM). Our findings are summarized in Table 2, while a detailed performance overview across FL training is shown in Figure 3.

The evaluation of defensive strategies against LSB, WaNet, Encoder, and BadNet attacks in FL reveals varied effectiveness. For LSB, neither Median, Trimmed Mean, nor FoolsGold consistently mitigate the attacks, with Trimmed Mean inadvertently increasing ASR by trimming honest clients and FoolsGold showing random, inconsistent mitigation. KRUM and Multi-KRUM show noticeable effects on ASR by selecting particular updates for aggregation, but KRUM often fails and Multi-KRUM significantly impacts model accuracy. Similar trends are observed for WaNet and Encoder attacks, with KRUM failing more often for the Encoder attack. In the case of BadNet, KRUM and Multi-KRUM provide some mitigation but are not consistently effective, with attacks persisting longer when attackers are chosen. Overall, these results suggest that attacker updates are not significantly different from other clients’ updates, leading to the “poison tangling effect” [14], where backdoor updates become ingrained in the model and indistinguishable from honest updates, making consistent mitigation challenging while significantly impacting model accuracy.

5.3 Improving FIBAs via ASR Boosting Techniques

In this section, we investigate *whether existing BAs boosting techniques, namely distribution of trigger across attackers, model replacement, or our proposed trigger hiding strategy in low model activation regions, can improve FiBAs ASR and/or stealthiness*. To this end, we performed experiments in CIFAR-10 under non-i.i.d. for $N=10$ and $m=0.1$, where we compared the performance of the original attack with its “boosted” variation under identical settings.

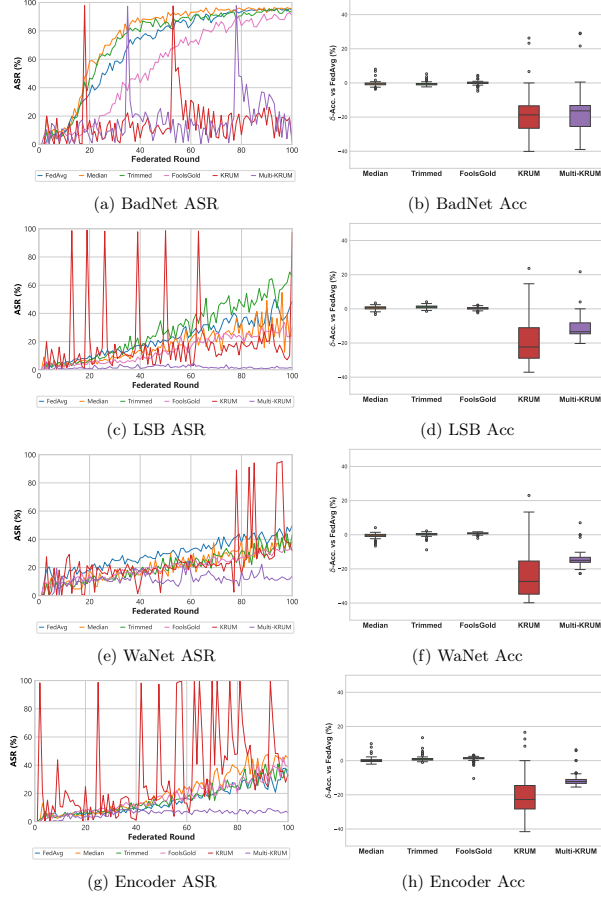


Fig. 3: Evaluation of FiBAs with $m=0.1$ against Defense Mechanisms for BAs on CIFAR-10. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

Distributed Triggers. To analyze the effect of trigger distribution on FiBAs, we apply the trigger distribution strategy explained in Section 3.1 to the LSB and WaNet attacks. The results, presented in Figure 4, show that distributing the trigger across clients has a negative effect on the ASR. Both attacks show almost no increase in ASR during the first 60 rounds. After round 60 the distributed LSB attack starts to show ASR improvements but the attack becomes very unstable. WaNet still shows no improvement after round 60. Splitting the trigger into smaller pieces makes local training too difficult for the already hard to learn triggers. This causes poor model updates in which the trigger is not properly learned, leading to a poor ASR after aggregation. Clearly, our current version of distributing triggers across clients is not effective at increasing the ASR of FiBAs. Unless a strategy is devised which optimizes the distribution of the triggers, similar to CBA [11], it will likely remain true that splitting up already hard to learn triggers will remain detrimental for FiBAs.

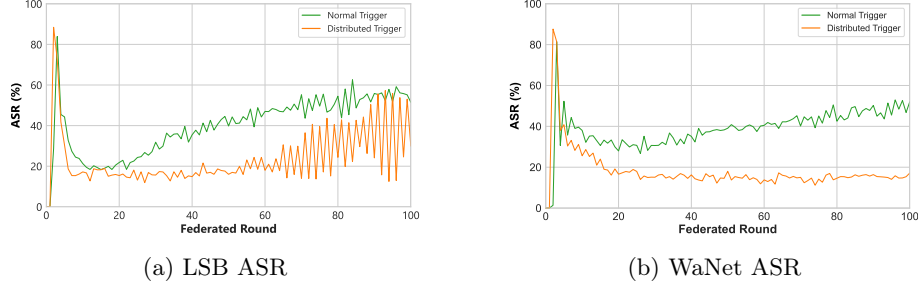


Fig. 4: Comparison of normal vs distributed trigger ($\times 4$) for LSB and WaNet attacks in CIFAR-10 with $m=0.1$. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

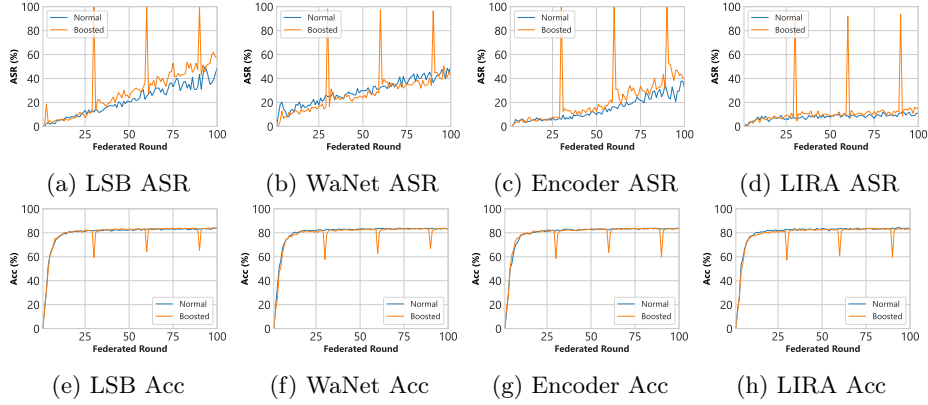


Fig. 5: Model Replacement in various FiBAs, where we report ASR and Acc with $m=0.1$. Parameters are set to $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

Model Replacement. We now analyze the effect of model replacement across FiBAs, where the model replacement strategy is *activated* in rounds 30, 60 and 90. From our results, presented in Figure 5, highlight peaks in ASR in those rounds (Figure 5a-d). From this, we derive that model replacement strategy is effective in replacing the global model by up-scaling the attacker model. However, in all cases, this ASR boosting diminishes in subsequent rounds. This is another indication of the heavy competition these attacks suffer from other clients. In terms of model performance (Figure 5e-h) we notice a sharp drop in accuracy in the *activated* rounds. As the global model is effectively replaced by the attacker’s model, the resulting model exhibits poor generalization. To conclude, model replacement can only increase FiBAs ASR in the current *activation* round, while in subsequent rounds its effects are quickly averaged out. Even if applied in the final round, this results in a poorly performing model.

We further investigate what the effect of simple outlier removal defense mechanism, such as Median or Trimmed Mean, when model replacement strategy is applied. From the results presented in Figure 6, we notice that while in standard FedAvg the ASR peaks are present in all 3 *activation* rounds (30, 60 and 90), these peaks are missing in both experiments where Median and Trimmed Mean

is applied during server-side aggregation. This is due to the fact that unlike regular model updates, scaled updates have significantly higher magnitudes over honest clients; thus easily distinguishable with outlier aggregation techniques.

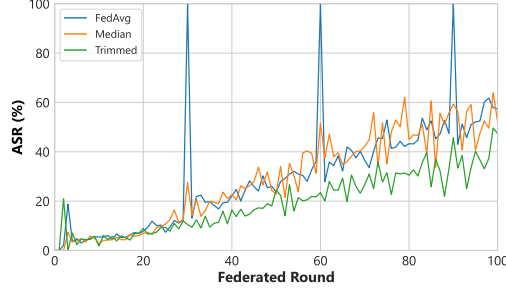


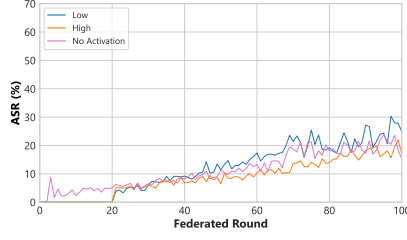
Fig. 6: Model Replacement under robust aggregation defence mechanisms in CIFAR-10 with $m=0.1$. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

Low Activation Triggers. Here we investigate whether our proposed strategy for increasing the ASR of FiBAs attacks by hiding the trigger in low activation areas has any merit. We use LSB_{500} and LSB_{700} to see if this strategy also provides improvements for smaller trigger sizes. To better show the effect of this strategy we apply the mask in two different settings, one where we use the mask to indicate low attention areas and one setting where we invert the mask to indicate the high attention areas. We further perform identical experiments, where no modifications are performed using the activation mask. We report the Neural Cleanse [31] anomaly scores to investigate whether our proposed trigger hiding strategy produces results in triggers which are easier to detect. Figure 7 shows the result of using the activation mask strategy for the LSB attack for the described settings.

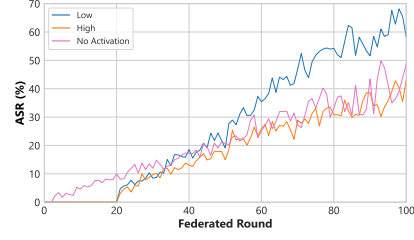
Figure 7 shows a strong difference between the high and low activation areas. The ASR in the low setting climbs much faster and reaches $\pm 25\%$ higher ASR for LSB_{700} . This highlights that there is a noticeable effect when avoiding regions other clients focus on. We see only a small increase for LSB_{500} , in this setting the

Method	Acc	ASR	LPIPS($\downarrow$) [33]	Anomaly Index($\downarrow$) [31]
No Act.	82.92 ± 0.25	20.17 ± 2.84	3.16×10^{-5}	1.79
LSB_{500} Low	83.79 ± 0.29	24.72 ± 3.51	4.25×10^{-5}	2.11
LSB_{500} High	83.40 ± 0.30	18.30 ± 1.94	7.05×10^{-5}	1.36
No Act.	83.37 ± 0.33	41.59 ± 6.19	7.83×10^{-5}	1.61
LSB_{700} Low	83.82 ± 0.34	61.39 ± 4.47	2.25×10^{-4}	3.04
LSB_{700} High	83.37 ± 0.31	36.89 ± 4.26	3.75×10^{-4}	2.01

Table 3: Effect of trigger hiding based on model activation regions (Low & High) in CIFAR-10 for LSB with $m=0.1$. For trigger hidings we use $LSB_{500,700}$, while No Act. refer to standard LSB attack. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.



(a) ASR for LSB_{500}



(b) ASR for LSB_{700}

Fig. 7: Effect of trigger hiding based on model activations regions (Low & High) in CIFAR-10 for LSB with $m=0.1$. Trigger sizes are LSB_{500} and LSB_{700} , No Activation refers to standard LSB attack. Parameters are $N = 10$, $R=100$, $E=5$, $\sigma=0.5$, and $t = 5$.

trigger seems to be too small in general. The low attention run also climbs faster, taking into account the attack starts 20 rounds later. This is likely due to the fact that the trigger faces less competition during the aggregation process causing it to climb faster. Next we observe the Neural Cleanse anomaly score. While both No Activation settings and the high activation setting mostly stay below the anomaly threshold of 2, the low attention setting causes a significant increase in the anomaly score. This indicates that, even though our proposed strategy is effective at increasing the ASR of the attack, by evading high attention regions it also made itself more obvious since it cannot 'hide' behind activation's in these regions. This makes it easier for a model inspection defense like Neural Cleanse to recover the trigger. Note that the high activation setting has comparable performance to the normal setting. It is likely that both the high and low settings achieve benefits over the normal setting because bits are flipped more often since there are less pixels the attack can flip in these cases, creating a higher trigger intensity. However the difference between high and low should indicate that this way of embedding the trigger has promise.

6 Conclusions

In this study, we explored the performance of invisible BA in FL. Our results show that invisible BA, while less effective than obvious ones like BadNet, can achieve reasonable ASR with over 20% of clients as attackers, and still perform adequately with only 10% attackers. Common FL defenses generally fail to mitigate these attacks without significantly degrading model accuracy. Techniques like model replacement and distributed attacks did not enhance ASR for invisible BA. However, hiding triggers in low attention regions improved ASR. Future work should optimize these strategies, compare low and high attention region approaches, and refine model replacement and distributed attack methods for better FL performance.

Acknowledgement

This research is partially funded by the DAIS project, which has received funding from KDTJU under grant agreement No 101007273.

References

1. Awan, S., Luo, B., Li, F.: Contra: Defending against poisoning attacks in federated learning. In: *Computer Security – ESORICS 2021*. pp. 455–475. Springer International Publishing, Cham (2021)
2. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning (2019)
3. Baluja, S.: Hiding images in plain sight: Deep steganography. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*. vol. 30. Curran Associates, Inc. (2017)
4. Beutel, D.J., Topal, T., Mathur, A., Qiu, X., Fernandez-Marques, J., Gao, Y., Sani, L., Li, K.H., Parcollet, T., de Gusmão, P.P.B., et al.: Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390* (2020)
5. Blanchard, P., Mhamdi, E.M.E., Guerraoui, R., Stainer, J.: Byzantine-tolerant machine learning (2017)
6. Chen, X., Liu, C., Li, B., Lu, K., Song, D.: Targeted backdoor attacks on deep learning systems using data poisoning (2017)
7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018)
8. Doan, K., Lao, Y., Zhao, W., Li, P.: Lira: Learnable, imperceptible and robust backdoor attacks. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. pp. 11946–11956 (2021). <https://doi.org/10.1109/ICCV48922.2021.01175>
9. Fung, C., Yoon, C.J.M., Beschastnikh, I.: The limitations of federated learning in sybil settings. In: *International Symposium on Recent Advances in Intrusion Detection* (2020)
10. Garrido, Q., Balestrieri, R., Najman, L., LeCun, Y.: Rankme: assessing the downstream performance of pretrained self-supervised representations by their rank. In: *Proceedings of the 40th International Conference on Machine Learning* (2023)
11. Gong, X., Chen, Y., Huang, H., Liao, Y., Wang, S., Wang, Q.: Coordinated backdoor attacks against federated learning with model-dependent triggers. *IEEE Network* **36**(1), 84–90 (2022). <https://doi.org/10.1109/MNET.011.2000783>
12. Gu, T., Liu, K., Dolan-Gavitt, B., Garg, S.: Badnets: Evaluating backdoor-ing attacks on deep neural networks. *IEEE Access* **7**, 47230–47244 (2019). <https://doi.org/10.1109/ACCESS.2019.2909068>
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
14. Huang, T., Hu, S., Chow, K.H., Ilhan, F., Tekin, S., Liu, L.: Lockdown: Backdoor defense for federated learning with isolated subspace training. In: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *Advances in Neural Information Processing Systems*. vol. 36, pp. 10876–10896. Curran Associates, Inc. (2023)
15. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
16. Li, S., Xue, M., Zhao, B.Z.H., Zhu, H., Zhang, X.: Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing* **18**(5), 2088–2105 (2021). <https://doi.org/10.1109/TDSC.2020.3021407>

17. Li, Y., Li, Y., Wu, B., Li, L., He, R., Lyu, S.: Invisible backdoor attack with sample-specific triggers (2021)
18. Liu, P., Xu, X., Wang, W.: Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives. *Cybersecurity* **5**, 4 (02 2022). <https://doi.org/10.1186/s42400-021-00105-6>
19. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data (2023)
20. Nadgauda, S.S., Pennamada, Y.R., D, S.: Steganet: A deep learning model for image steganography using customized cnn and autoencoders. In: 2023 OITS International Conference on Information Technology (OCIT). pp. 196–201 (2023). <https://doi.org/10.1109/OCIT59427.2023.10431390>
21. Nguyen, A., Tran, A.: Wanet – imperceptible warping-based backdoor attack (2021)
22. Nguyen, T.D., Nguyen, T., Nguyen, P.L., Pham, H.H., Doan, K., Wong, K.S.: Backdoor attacks and defenses in federated learning: Survey, challenges and future research directions (Mar 2023)
23. Omeiza, D., Speakman, S., Cintas, C., Weldermariam, K.: Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models (2019)
24. Provos, N.: Defending against statistical steganalysis. 10th USENIX Security Symposium (2001)
25. Rodríguez-Barroso, N., Jiménez-López, D., Luzón, M.V., Herrera, F., Martínez-Cámara, E.: Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges. *Information Fusion* **90**, 148–173 (feb 2023). <https://doi.org/10.1016/j.inffus.2022.09.011>
26. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision* **128**(2), 336–359 (Oct 2019). <https://doi.org/10.1007/s11263-019-01228-7>
27. Shi, Y., Song, H., Xu, J.: Responsible and effective federated learning in financial services: A comprehensive survey. In: 2023 62nd IEEE Conference on Decision and Control (CDC). pp. 4229–4236 (2023). <https://doi.org/10.1109/CDC49753.2023.10384119>
28. Subramanian, N., Cheheb, I., Elharrouss, O., Al-Maadeed, S., Bouridane, A.: End-to-end image steganography using deep convolutional autoencoders. *IEEE Access* **9**, 135585–135593 (2021). <https://doi.org/10.1109/ACCESS.2021.3113953>
29. Subramanian, N., Elharrouss, O., Al-ma'adeed, S., El-Seoud, S.: Image Steganography Using Auto Encoder-Decoder Based Deep Learning Method, pp. 520–530 (02 2021). https://doi.org/10.1007/978-3-030-67209-6_56
30. Tolpegin, V., Truex, S., Gursoy, M.E., Liu, L.: Data poisoning attacks against federated learning systems (2020)
31. Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., Zhao, B.Y.: Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In: 2019 IEEE Symposium on Security and Privacy (SP). pp. 707–723 (2019). <https://doi.org/10.1109/SP.2019.00031>
32. Zhang, L., Zheng, B.: Fiba: Federated invisible backdoor attack. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6870–6874 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10446910>
33. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (2018)