MARS: A Malignity-Aware Backdoor Defense in Federated Learning

Wei Wan[1,‡,§], **Yuxuan Ning**[2,‡], **Zhicong Huang**[3], **Cheng Hong**[3], **Shengshan Hu**[4], **Ziqi Zhou**[5,*], **Yechao Zhang**[6], **Tianqing Zhu**[1], **Wanlei Zhou**[1], **Leo Yu Zhang**[7]
[1] Faculty of Data Science, City University of Macau
[2] School of Computing, Australian National University [3] Ant Group
[4] School of Cyber Science and Engineering, Huazhong University of Science and Technology
[5] School of Computer Science and Technology, Huazhong University of Science and Technology
[6] College of Computing and Data Science, Nanyang Technological University
[7] School of Information and Communication Technology, Griffith University
{weiwan,tqzhu,wlzhou}@cityu.edu.mo Yuxuan.Ning@anu.edu.au
{zhicong.hzc,vince.hc}@antgroup.com {zhouziqi,hushengshan}@hust.edu.cn
yech.zhang@gmail.com leo.zhang@griffith.edu.au

Abstract

Federated Learning (FL) is a distributed paradigm aimed at protecting participant data privacy by exchanging model parameters to achieve high-quality model training. However, this distributed nature also makes FL highly vulnerable to backdoor attacks. Notably, the recently proposed state-of-the-art (SOTA) attack, 3DFed (SP2023), uses an indicator mechanism to determine whether the backdoor models have been accepted by the defender and adaptively optimizes backdoor models, rendering existing defenses ineffective. In this paper, we first reveal that the failure of existing defenses lies in the employment of empirical statistical measures that are loosely coupled with backdoor attacks. Motivated by this, we propose a **M**alignity-**A**ware backdoo**R** defen**S**e (MARS) that leverages backdoor energy (BE) to indicate the malicious extent of each neuron. To amplify malignity, we further extract the most prominent BE values from each model to form a concentrated backdoor energy (CBE). Finally, a novel Wasserstein distance-based clustering method is introduced to effectively identify backdoor models. Extensive experiments demonstrate that MARS can defend against SOTA backdoor attacks and significantly outperforms existing defenses.

1 Introduction

Federated Learning (FL) [26, 33, 29, 23] is a distributed machine learning paradigm that leverages data distributed across multiple clients to train a high-quality global model without requiring data to be shared with a third party. Due to its exceptional privacy-preserving features and efficient utilization of decentralized data, FL has found widespread applications in fields such as healthcare [1], education [10], finance [7], and even the military [2]. However, the distributed nature also makes it highly susceptible to poisoning attacks [31, 36, 20, 24, 18]. Among these, Byzantine attacks aim to degrade the global model accuracy, while backdoor attacks trigger malicious behavior (*e.g.*, classify any input as the attacker's desired target class) only under specific conditions (*e.g.*, a white patch in the bottom right corner of an image). Because backdoor attacks do not affect the model's performance

[‡]These authors contributed equally to this work.
[§]Work done during an internship at Ant Group.
[*]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

on clean samples, it is difficult for model users to realize that a backdoor has been implanted [45, 39]. This makes backdoor attacks a greater potential threat to FL.

To defend against backdoor attacks, the FL community has made significant efforts. Certain defenses constrain the norm of local updates to prevent backdoor updates from dominating the global model [35, 38, 6, 34]. Other strategies employ out-of-distribution (OOD) detection techniques to eliminate local updates that significantly deviate from the overall distribution [27, 43, 40, 5]. Additionally, some defenses focus on detecting model consistency, such as the cosine similarity of updates, and assign lower aggregation weights to updates with high consistency (indicative of Sybil attacks) or remove them altogether [12, 30]. However, these defenses offer limited protection. Recently proposed state-of-the-art (SOTA) attacks can easily bypass these measures. For instance, 3DFed [17] uses an indicator mechanism to determine if backdoor updates are being aggregated, allowing for adaptive optimization of local models. DarkFed [19] and CerP [25] introduce several constraint terms that make backdoor updates resemble benign updates, exhibiting properties such as moderate magnitude, reasonable distribution, and limited consistency, making it difficult to distinguish between benign and backdoor updates. These sophisticated attacks pose a significant threat to the security of FL, underscoring the urgent need for effective defenses.

In this paper, we first reveal through experimental observations that the primary statistical measures relied upon by existing defenses fail to distinguish between benign and backdoor updates when faced with SOTA attacks. We attribute this failure to the fundamental reason that ***these statistical measures are empirical and loosely coupled with backdoor attacks.*** In other words, these statistical metrics do not inherently reflect whether a local update has been compromised with a backdoor. The lack of perceiving malicious intent in existing defenses provides attackers with the opportunity to mimic the statistical distribution of benign updates, thereby defeating these defenses. Motivated by this, we propose MARS, a **M**alignity-**A**ware backdoo**R** defen**S**e. Specifically, we introduce the concept of backdoor energy (BE), which indicates the malignancy level of each neuron in the model (*i.e.*, its relevance to backdoor intent), thereby achieving a strong coupling with backdoor attacks. To amplify the malignity, we further extract the most prominent BE values in each local model to form the concentrated backdoor energy (CBE), concentrating the backdoor information. Finally, a novel Wasserstein distance-based clustering algorithm is proposed to detect backdoor models. This new clustering focuses on the probability density of elements in CBEs, thus avoiding the issues of element order sensitivity encountered by existing Euclidean and cosine distance-based clustering methods. An overview of MARS is illustrated in Figure 1.

In summary, the contributions of this paper are as follows:

- We identify the failure of existing FL backdoor defenses, attributing their failures to a reliance on empirical statistical measures that are loosely coupled with backdoor attacks. From a new perspective, we propose a robust FL defense strategy with malignity-aware capabilities.
- We introduce MARS, which detects potentially harmful neurons by incorporating the concept of backdoor energy, and we also propose a Wasserstein distance-based clustering algorithm to enhance the precise identification of backdoor models.
- We conduct extensive experiments to evaluate the effectiveness of MARS. The results demonstrate that MARS can counter SOTA backdoor attacks and consistently provide superior protection for FL compared to existing defenses.

2 Related Work

2.1 Backdoor Attacks in Federated Learning

Since its inception, FL has been a focal point for research on backdoor vulnerabilities. Model Replacement Attack (MRA) [3], the pioneering backdoor attack on FL, works by proportionally amplifying backdoor updates to ensure that even a few malicious updates can dominate the global model. Wang et al. [38] later introduced the edge-case backdoor attack, leveraging rare samples from the dataset's tail to activate backdoors. Xie et al. [41] proposed DBA, which divides a complete trigger into multiple sub-triggers assigned to different attackers to create backdoor samples, aiming to reduce the pairwise similarity of malicious updates. However, these early attacks often neglected

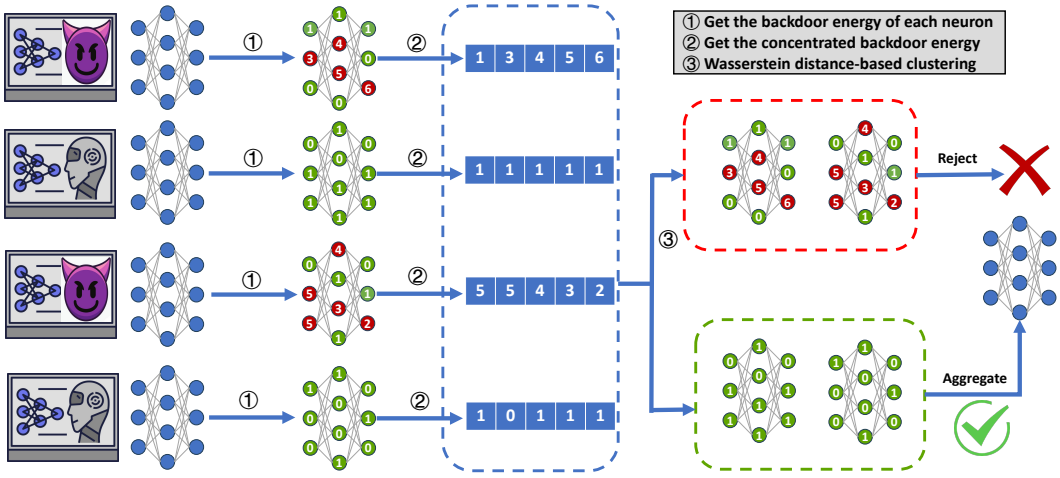


Figure 1: Overview of MARS. To facilitate understanding, we simplify the FL system to include only four clients. The first and third clients upload backdoor models, while the second and fourth clients upload benign models. Red circles represent higher backdoor energy, while green circles indicate lower backdoor energy.

the possibility of defensive measures, rendering them effective primarily against FL systems with no defenses or only weak ones.

To address this shortcoming, a new wave of sophisticated backdoor attacks has emerged. 3DFed [17], for instance, employs an indicator mechanism to detect whether backdoor updates are being aggregated and then adaptively optimizes the backdoor models to evade existing defenses. Similarly, CerP [25] and DarkFed [19] share a core strategy of adding constraints to mimic the characteristics of benign updates—such as moderate magnitudes and limited consistency—striking a balance between stealth and efficacy. These advanced attacks significantly threaten the secure deployment of FL, necessitating the development of robust defenses.

2.2 Backdoor Defenses in FL

We broadly categorize existing defenses into three main types based on the techniques they employ: norm constraint-based defenses, OOD detection-based defenses, and consistency detection-based defenses.

Norm constraint-based defenses posit that the optimal point for the backdoor task typically deviates significantly from the optimal point for the main task. This results in the norm of backdoor updates being much larger than that of benign updates. Consequently, these defenses constrain the norm of all local updates within a reasonable range. Norm Clipping [38] serves as a representative example of such defenses. Additionally, some other defenses [35, 34, 6] also leverage this characteristic to prevent malicious updates from dominating the global model.

OOD detection-based defenses assert that backdoor updates and benign updates exhibit substantial differences in their distributions, with benign updates typically being densely distributed. In contrast, backdoor updates can be considered as outliers. Building on this premise, Multi-Krum [5] calculates an anomaly score for each local update based on the sum of its distances to its neighboring nodes. A higher score indicates greater deviation, making it more likely to be discarded. RFLBAT [40] utilizes Principal Component Analysis (PCA) to project local updates into a low-dimensional space. Subsequently, it employs a clustering algorithm to identify outliers, marking them as backdoor updates. FLAME [27] identifies updates that deviate significantly in direction from the overall trend as backdoor updates and excludes them from the aggregation queue. FLDetector [43] exploits the differences between the predicted model and the actual model to discover outliers.

Consistency detection-based defenses argue that all backdoor updates share the same objective, namely, to classify trigger-carrying samples as the target label. Therefore, these updates exhibit strong consistency, either in terms of update directions or neuron activations. On the other hand, diverse benign updates may display lower consistency due to data heterogeneity [22]. With this

understanding, FoolsGold [12] assigns lower aggregation weights to updates with high pairwise cosine similarities, thereby mitigating the impact of backdoor updates. DeepSight [30] uses the consistency on neuron activations in the backdoor model to detect malicious updates.

3 Threat Model

3.1 Attack Model

The primary objective of the attackers is to implant a backdoor into the global model by transmitting malicious model parameters to the central server. To facilitate more sophisticated backdoor attacks, we assume the attackers possess substantial capabilities:

- **Flexible Local Optimization.** Attackers can arbitrarily modify their local optimization objectives, achieving a fine balance between stealth and effectiveness.
- **Collusion Capability.** Attackers can collude, allowing full transparency of training data and model parameters among them. This transparency aids in dynamically adjusting backdoor models to evade defense mechanisms.
- **Dominant Presence.** Attackers can constitute a majority, with their proportion not restricted to below 50% as typically assumed in existing research.

These powerful assumptions significantly heighten the challenge of defending against backdoor attacks.

3.2 Defense Model

Our proposed defense is deployed at the central server to detect and filter out backdoor models from the local models uploaded by clients, resulting in a high-performance, backdoor-free global model. We assume the central server has minimal knowledge. Specifically, the server only has access to the model parameters of all local models in each round. It cannot access any client's training data or control the training process of any client's model. Moreover, the server does not make any assumptions about the proportion of attackers. Our proposed defense algorithm aims to achieve the following goals simultaneously:

- **Effectiveness.** Regardless of the type of backdoor attack, the defense should effectively thwart the attackers' malicious activities, resulting in a backdoor-free global model.
- **Practicability.** The defense should remain effective in challenging real-world scenarios, such as when the proportion of attackers exceeds 50%, clients have heterogeneous data distributions, or clean auxiliary datasets are unavailable.
- **Fidelity.** In non-adversarial scenarios (*i.e.*, there are no attackers in the FL system), the accuracy of the global model on clean samples should not degrade compared to FedAvg due to the deployment of this defense.

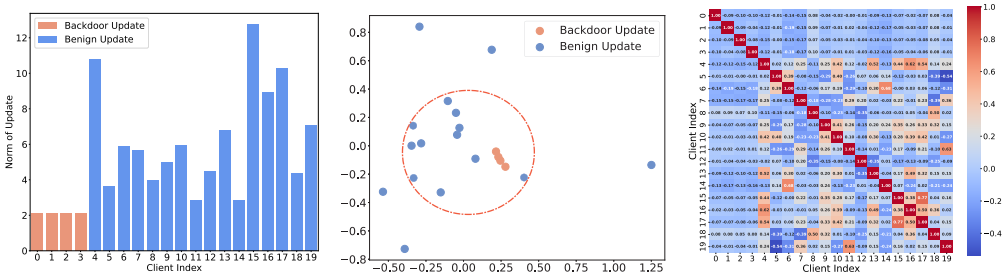
These objectives ensure that the defense is robust, practical, and reliable in both adversarial and non-adversarial environments.

4 MARS

4.1 Motivation

After reviewing the SOTA defenses, we find that they mainly rely on empirical statistical measures. Techniques such as norm constraint, OOD detection, and consistency detection are extensively utilized by them. However, we demonstrate that these empirical statistical measures tend to fail when faced with advanced attacks.

Failure of Norm Constraint. To prevent the norm (also known as magnitude) of backdoor updates from becoming excessively large, some advanced backdoor attacks [19, 17, 25] incorporate a constraint term to encourage finding backdoor models near the previous round's global model. The resulting backdoor updates, even without proportionally increasing their magnitude, can still achieve excellent attack efficacy. As shown in Figure 2(a), the magnitude of backdoor updates obtained this way can be even smaller than that of benign updates. This indicates that when a defender employs the



(a) Norm Comparison (b) Distribution Comparison (c) Consistency Comparison

Figure 2: Comparison of statistical measures between backdoor and benign updates. We consider 20 clients, with the first 4 clients (indices 0 to 3) being malicious and conducting 3DFed attack, while the remaining clients (indices 4 to 19) are benign. (a) provides the norms of all local updates. (b) shows the distribution of all local updates projected into 2D space using PCA. (c) presents a heatmap of the similarities between local updates.

norm constraint, backdoor updates remain unaffected. Consequently, this type of statistical measure can be easily bypassed by these advanced backdoor attacks.

Failure of OOD Detection. To make backdoor updates appear less anomalous, several advanced backdoor attacks have devised innovative solutions. 3DFed [17] generates a series of outlier decoy updates, making the backdoor updates seem more benign in comparison, thus bypassing the detection by a defender. DarkFed [19] adds a constraint term to ensure that the cosine similarity between backdoor and benign updates is close to that among benign updates themselves. CerP [25] employs a similar strategy to DarkFed but uses Euclidean distance for constraint. As shown in Figure 2(b), the backdoor updates crafted by 3DFed are indistinguishable from benign ones when projected onto a 2-dimensional space. Moreover, Figure 2(c) provides new evidence from another perspective. It illustrates that the cosine similarity between backdoor and benign updates hovers around -0.08, which is even higher than the similarity among some benign updates (*e.g.*, a cosine similarity of -0.54 between client 5 and client 19). Consequently, OOD detection fails to provide effective protection against these attacks.

Failure of Consistency Detection. To reduce the consistency of backdoor updates, 3DFed [17] adds carefully designed noise masks to each backdoor update, increasing the variability among them without diminishing the strength of the attack. DarkFed [19] and CerP [25] achieve a similar effect by adding a constraint term to decrease the cosine similarity between pairs of backdoor updates. As shown in Figure 2(c), the cosine similarity between backdoor updates is only about -0.08, which is significantly lower than the cosine similarity among some benign updates (*e.g.*, 0.77 between client 15 and client 17). This indicates that consistency detection also fails to differentiate between backdoor and benign updates.

We attribute all the aforementioned failures to a fundamental reason: ***these empirical statistical measures are loosely coupled with backdoor attacks.*** In other words, they lack the capability to perceive malicious intent and do not fundamentally reflect whether an update has been compromised with a backdoor. Consequently, attackers can easily mimic the statistical measures of benign updates, thereby bypassing existing defenses. This motivates us to employ a malignity-aware measure that can reflect the inherent maliciousness of the model, rather than relying on empirical intuitions.

4.2 Overview of MARS

Unlike existing schemes that directly detect abnormal statistical measures based on model parameters, we propose a **M**alignity-**A**ware backdoo**R** defen**S**e (MARS). As shown in Figure 1, for each local model, we first calculate the *backdoor energy* (BE) of each neuron, which reflects how strongly a neuron is associated with backdoor attacks. Higher backdoor energy indicates a higher level of malignity for that neuron. To further amplify the malignity, we extract the most prominent backdoor energies from each layer and concatenate them into a one-dimensional vector, which we call the *concentrated backdoor energy* (CBE). Note that CBE is not unique to backdoor models; it can also be calculated for benign models. Subsequently, we propose a novel Wasserstein-based clustering method to effectively identify backdoor models and prevent them from participating in the aggregation.

4.3 Obtaining Backdoor Energy

Given an L-layer neural network $F^* = f^{(L)} \circ f^{(L-1)} \circ \ldots \circ f^{(1)}$, a clean dataset $D \subseteq \mathcal{X} \times \mathcal{Y}$, and a backdoor trigger generator $\delta(.)$, a straightforward way to evaluate the backdoor energy of the k^{th} neuron in the l^{th} layer is to compute the expected difference in neuron values between clean samples and backdoor samples:

$$BE_k^{(l)}(F) = \mathbb{E}_{x \sim \mathcal{X}} \left\| F_k^{(l)}(x) - F_k^{(l)}(\delta(x)) \right\|_2, \tag{1}$$

where $F_k^{(l)}(.) = f_k^{(l)} \circ f^{(l-1)} \circ \ldots \circ f^{(1)}(.)$ indicates the neuron function that maps an input sample to the k^{th} neuron in the l^{th} layer.

However, obtaining BE via equation 1 faces two challenges. First, due to the privacy-preserving nature of FL, the clean dataset D is inaccessible to the defender. Furthermore, the trigger is very subtle and private to the attackers, making it difficult for the defender to obtain. A naive idea is to collect a shadow dataset and employ reverse engineering [37] to reconstruct the trigger. However, the shadow dataset may significantly different from the real training dataset, leading to inaccurate BE calculations and impairing the detection of backdoor models. Moreover, reverse engineering requires reconstructing a trigger for each class individually, which becomes very time-consuming when there are many classes (*e.g.*, ImageNet with 1000 classes). Additionally, when the trigger is complex, the reconstructed trigger may significantly differ from the real one, also resulting in inaccurate BE calculations. Considering the above challenges, we turn to exploring the upper bound of BE.

Theorem 4.1 (**Upper Bound of Backdoor Energy**). *Suppose an L-layer neural network F and its every sub-network $f^{(l)}, l \in [1, L]$, are Lipschitz smooth. Then, the backdoor energy of the k^{th} neuron in the l^{th} layer can be upper bounded by:*

$$BE_k^{(l)}(F) \leq \|f_k^{(l)}\|_{Lip} \prod_{i=1}^{l-1} \|f^{(i)}\|_{Lip} \mathbb{E}_{x \sim \mathcal{X}} \|x - \delta(x)\|_2, \tag{2}$$

where $\|.\|_{Lip}$ represents the Lipschitz constant of a function. The detailed proof is provided in Appendix A.

On the one hand, we do not need the exact value of BE for subsequent calculations, but only the relative magnitudes of BE among different neurons to detect anomalies. On the other hand, the upper bound of backdoor energy reasonably reflects the distribution of BE. Thus we can approximate BE using its upper bound. Furthermore, as indicated by formula 2, when considering different neurons j and k in the same layer l, the difference in their BE upper bounds is solely in the first term, *i.e.*, $\|f_j^{(l)}\|_{Lip}$ and $\|f_k^{(l)}\|_{Lip}$. Therefore, we can further approximate BE using only the first term of its upper bound:

$$BE_k^{(l)}(F) = \|f_k^{(l)}\|_{Lip}. \tag{3}$$

Notably, equation 3 does not rely on the clean dataset or the trigger. It allows for the easy calculation of BE for all neurons using only the model parameters. Equation 3 aligns with the empirical findings reported in CLP [44]. The primary distinctions between MARS and CLP are detailed in Appendix B. Appendix C provides more details about the calculation of Lipschitz constant.

4.4 Obtaining Concentrated Backdoor Energy

Since a backdoor can be viewed as a shortcut [37], only a small number of neurons are backdoor-related. Therefore, we extract the highest BE values from each layer (*e.g.*, the top 5% by default in our paper) and concatenate them into a one-dimensional vector. We call this vector the concentrated backdoor energy (CBE), as it aggregates the most prominent backdoor energies in the model. This approach minimizes interference from neurons unrelated to the backdoor, aiding in the subsequent differentiation between backdoor and benign models. Formally, the CBE of a model F can be obtained by:

$$CBE(F) = \bigcup_{l=1}^{L} TopK_{\kappa\%} \left(\{BE_i^{(l)}(F)\}_{i=1}^{n_l} \right), \tag{4}$$

*We omit the model weights θ in $F(.;\theta)$ for simplicity.

where L is the total number of layers, n_l is the number of neurons in the l^{th} layer, $TopK_{\kappa\%}(.)$ denotes the top $\kappa\%$ values of a set.

4.5 Identifying Backdoor Models

CBE can effectively capture each local model's backdoor information, in which backdoor and benign models are quite different, making clustering a promising approach for identifying backdoor models. However, two challenges remain to be addressed. First, existing clustering methods primarily use Euclidean distance or cosine distance as metrics, which are highly sensitive to the order of elements rather than their overall distribution, leading to potential errors. Second, after clustering, it is challenging to decide which clusters to trust and include in the final aggregation. Choosing the wrong clusters could result in the failure to exclude backdoor models. Compulsively discarding some clusters, in innocent scenario (*i.e.*, all clients are benign), may slow down the convergence of the global model or even decrease its accuracy.

Wasserstein Distance-Based Clustering. We use K-Means to partition the CBEs of all local models into two clusters. However, the default metric, Euclidean distance, or the widely used cosine distance[†], is sensitive to the order of elements and not suitable for our scenario. This is particularly true for FL, where the top BE values of different local backdoor models may appear in different neurons due to the distributed nature of training. As a result, even though the elements in the CBEs of backdoor models are generally larger, both Euclidean and cosine distances do not recognize these CBEs as similar. To focus on the distribution of elements in the CBE and avoid the influence of their order, we employ Wasserstein distance [28] as the metric for K-Means and call the clustering algorithm K-WMeans. Formally, for two probability distributions p and q, the Wasserstein distance between them is defined as:

$$Wass(p,q) = \inf_{\gamma \sim \prod(p,q)} \mathbb{E}_{x,y\sim\gamma}\|x-y\|, \tag{5}$$

where $\prod(p,q)$ denotes the set of all possible joint distributions between p and q. Next, we use a toy example to demonstrate that Wasserstein distance is more suitable than Euclidean and cosine distances for identifying backdoor models in our case.

Toy Example: *Assume $L1 = [1,2,3,4,5]$ and $L2 = [5,5,3,2,2]$ are the CBEs of backdoor models, and $L3 = [1,1,1,1,1]$ is the CBE of a benign model. This assumption is reasonable because neurons in backdoor models have higher BE values. As shown in Table 1, when considering Euclidean distance, $L1$ and $L3$ are deemed the closest. When using cosine distance as the metric, $L2$ and $L3$ are considered the closest. Both metrics are not conducive to clustering backdoor models into a single cluster. Notably, when considering Wasserstein distance, despite the significant differences in values across each dimension for $L1$ and $L2$, their distance is much smaller than the distances between $L1$ and $L3$ or $L2$ and $L3$. This favors the clustering of backdoor models together.*

Table 1: Metric comparison.

Metric	(L1, L2)	(L1, L3)	(L2, L3)
Euc.	6.16	**5.48**	6.16
Cos.	0.31	0.10	**0.07**
Wass.	**0.40**	2.00	2.40

Cluster Selection. After using K-WMeans to divide the CBEs into two clusters, the subsequent challenge is how to select the trusted cluster. Existing methods typically assume that benign clients are the majority and therefore accept the larger cluster. However, in some extreme scenarios, the number of attackers might exceed that of benign clients, leading to the unintended selection of backdoor models. To avoid this assumption, we use the norm of the cluster center as a more reliable metric for cluster selection. Specifically, the elements in the CBEs of attackers generally have higher values than those in the CBEs of benign clients. Therefore, we select the cluster with the smaller center norm, rather than relying on the majority.

However, when there are no attackers in the FL system, blindly discarding the cluster with the larger center norm might slow down the global model's convergence or even reduce its accuracy. To address this, we use the Wasserstein distance to measure the similarity between clusters. If the distance between the two clusters does not exceed a threshold ϵ, it indicates that the CBEs of all local models have similar distributions, corresponding to a scenario where all local models are either benign or malicious. Given that an FL system with only attackers is meaningless, we assume that when the cluster distance is low, all local models are benign. Therefore, in this case, both clusters are selected. A detailed algorithm description is provided in Appendix D.

[†]One minus cosine similarity.

5 Experiments

5.1 Experimental Setup

We consider an FL system with 100 clients, where 20 of them are designated as attackers. In each round, 20 clients are selected to participate in the FL process, with 4 of them guaranteed to be attackers. By default, MARS's hyperparameters κ and ϵ are set to 5 and 0.03, respectively. We fix the random seed to ensure reproduction and conduct experiments on the NVIDIA 3090Ti.

Datasets, models, and codes. We evaluate the effectiveness of MARS on MNIST [16], CIFAR-10 [15], and CIFAR-100 [15] datasets. To simulate realistic non-IID distributions, we use the Dirichlet distribution with a default sampling parameter α set to 0.9. For MNIST, a simple CNN is employed as the global model, while for CIFAR-10 and CIFAR-100, we use ResNet-18 [13] as the global model. The codes are available at https://github.com/yunming181920/MARS.

Evaluated attacks and defenses. We consider three SOTA backdoor attacks: MRA [3], CerP [25], and 3DFed [17]. Additionally, we design a customized adaptive attack tailored specifically for MARS. On the defense side, we evaluate eight SOTA defense methods, including FedAvg [26], Multi-Krum [5], RFLBAT [40], FLAME [27], FoolsGold [12], FLDetector [43], Deepsight [30], and FedCLP [44]. Notably, we also include the recently published backdoor defense, BackdoorIndicator [21], from Usenix Security 2024. Detailed descriptions of these attacks and defenses can be found in Appendix E.

Evaluation metrics. We assess the performance of the defenses using several metrics, including model accuracy (ACC), attack success rate (ASR), true positive rate (TPR), false positive rate (FPR), and comprehensive ability of defense (CAD). Higher values of ACC, TPR, and CAD, along with lower values of ASR and FPR, indicate a more effective defense. A more detailed description to these metrics can be found in Appendix F.

Table 2: Comparison of MARS and SOTA defenses under SOTA attacks.

Dataset	Baselines	MRA					CerP					3DFed				
		ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
MNIST	FedAvg	98.46	99.67	0.00	0.00	49.70	99.08	88.32	0.00	0.00	52.69	98.96	77.17	0.00	0.00	55.45
	Multi-Krum	98.97	9.73	100.00	0.00	97.31	99.34	9.76	100.00	0.00	97.40	99.08	85.06	0.00	25.00	47.26
	RFLBAT	98.98	9.71	100.00	19.38	92.47	90.84	21.00	90.00	16.88	85.74	99.13	74.53	0.00	18.75	51.46
	FLAME	98.98	9.63	100.00	31.25	89.53	99.31	9.74	100.00	31.25	89.58	98.65	91.41	0.00	56.25	37.75
	FoolsGold	99.00	9.64	100.00	0.00	97.34	99.31	29.21	30.00	0.00	75.03	99.02	73.08	0.00	0.00	56.49
	FLDetector	96.61	94.70	0.00	17.50	46.10	99.02	80.61	10.00	0.00	57.10	98.73	74.01	0.00	0.00	56.18
	DeepSight	98.92	22.83	0.00	6.25	67.46	99.29	9.75	100.00	37.50	88.01	98.74	62.30	0.00	6.25	57.55
	FedCLP	82.00	14.47	-	-	83.77	99.21	9.75	-	-	94.73	85.54	16.69	-	-	84.43
	MARS	99.01	9.66	100.00	0.00	97.34	99.32	9.74	100.00	0.00	97.40	99.13	9.72	100.00	3.62	96.45
CIFAR-10	FedAvg	78.32	99.68	0.00	0.00	44.66	84.49	93.70	0.00	0.00	47.70	84.37	96.76	0.00	0.00	46.90
	Multi-Krum	85.21	9.69	100.00	0.00	93.88	85.32	10.01	100.00	0.00	93.83	84.07	97.27	0.00	25.00	40.45
	RFLBAT	85.13	9.33	97.50	1.25	93.01	85.20	10.39	100.00	0.00	93.70	84.30	92.02	0.00	5.00	46.82
	FLAME	84.87	8.74	100.00	31.25	86.22	85.34	10.59	100.00	31.25	85.88	83.06	97.50	2.50	55.63	33.11
	FoolsGold	85.06	9.71	100.00	12.50	90.71	85.00	91.00	0.00	0.00	48.50	84.11	96.29	0.00	0.25	46.89
	FLDetector	85.16	9.96	100.00	0.00	93.80	85.18	88.64	50.00	0.00	62.39	84.24	95.20	0.00	35.00	38.51
	DeepSight	83.99	99.94	0.00	6.25	44.45	85.22	74.15	10.00	40.00	45.27	84.80	98.85	0.00	6.25	44.93
	FedCLP	75.01	10.88	-	-	82.07	78.52	11.00	-	-	83.76	69.25	7.55	-	-	80.85
	MARS	85.16	9.40	100.00	0.00	93.94	85.37	10.03	100.00	0.00	93.84	85.07	9.86	100.00	0.00	93.80
CIFAR-100	FedAvg	77.97	100.00	0.00	0.00	44.49	78.87	99.97	0.00	0.00	44.73	78.89	100.00	0.00	0.00	44.72
	Multi-Krum	79.34	0.97	100.00	0.00	94.59	79.67	1.14	100.00	0.00	94.63	78.36	100.00	0.00	25.00	38.34
	RFLBAT	79.46	0.97	100.00	15.00	90.89	79.50	1.15	100.00	0.63	94.43	78.89	100.00	0.00	18.75	40.04
	FLAME	79.63	0.95	100.00	31.25	86.86	79.56	1.20	100.00	31.25	86.78	79.27	1.00	100.00	31.25	87.76
	FoolsGold	79.54	0.98	100.00	0.00	94.64	79.59	1.16	100.00	0.00	94.61	79.01	100.00	0.00	0.00	44.75
	FLDetector	78.10	100.00	0.00	0.00	44.53	78.57	90.10	10.00	0.00	49.62	78.16	100.00	0.00	50.00	42.04
	DeepSight	78.85	61.30	0.00	6.25	52.83	79.18	1.20	97.50	56.88	79.65	78.91	10.59	20.00	25.00	65.83
	FedCLP	77.96	0.91	-	-	88.53	78.36	1.19	-	-	88.59	77.73	0.99	-	-	88.37
	MARS	79.53	0.97	100.00	0.00	94.64	79.73	1.15	100.00	0.00	94.65	79.37	0.97	100.00	0.00	94.60

5.2 Experimental Results

Comparison with SOTA defenses. Table 2 compares the performance of MARS with 8 SOTA defenses against 3 SOTA backdoor attacks across 5 evaluation metrics on 3 datasets. Overall, existing defenses fail to provide adequate protection, especially when confronted with advanced attacks like 3DFed. In contrast, our proposed MARS consistently achieves the best performance across all

datasets and attack scenarios, demonstrating its robustness in maintaining model performance in the presence of backdoor attacks. Specifically, for the MRA attack, defenses such as Multi-Krum, RFLBAT, FLAME, and FoolsGold achieve satisfactory ASR, but they suffer from excessive client exclusion. For instance, FLAME shows an FPR as high as 31.25% on the CIFAR-10 dataset. For the CerP attack, the effectiveness of existing defenses varies significantly across datasets. For example, FoolsGold can precisely detect all backdoor models on CIFAR-100, but only a few on MNIST, while its ability to detect backdoors completely breaks down on CIFAR-10. For 3DFed, most defenses show consistently high ASR, with the only exception being FedCLP, which achieves a relatively lower ASR, indicating some level of backdoor mitigation. However, FedCLP's aggressive pruning of local models often leads to excessive removal, negatively impacting the model's utility. When benchmarked against MARS, FedCLP's ACC drops by $1.64\% \sim 15.82\%$ across different datasets, a decline that is unacceptable for most real-world scenarios. We attribute MARS's superior defense ability to its detection of anomalies through BE, which is strongly correlated with backdoor attacks, fundamentally distinguishing it from existing defenses that rely on loosely coupled empirical metrics.

Resilience to adaptive attack. To further assess the robustness of MARS, we consider a more informed adversary, where attackers have prior knowledge that the central server employs MARS as the defense mechanism. Leveraging this insight, the attackers can craft adaptive strategies specifically designed to bypass MARS. Since MARS detects backdoor models through their relatively higher backdoor energy, a straightforward approach for executing an adaptive attack is to introduce a regularization term to minimize the backdoor energy of each neuron in each backdoor model. Formally, the attackers' optimization objective is defined as follow:

Table 3: Performance of MARS against adaptive attack.

λ	Defense	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
0.0001	**MARS**	85.31	9.43	100.00	0.00	93.97
	MARS*	85.45	9.86	100.00	0.00	93.90
0.001	**MARS**	85.18	9.75	100.00	0.00	93.86
	MARS*	85.05	9.44	100.00	0.00	93.90
0.01	**MARS**	85.26	9.57	100.00	0.00	93.92
	MARS*	85.50	9.76	100.00	0.00	93.94
0.05	**MARS**	10.00	100.00	0.00	97.50	3.13
	MARS*	85.12	9.30	100.00	0.00	93.96
0.1	**MARS**	10.00	100.00	0.00	99.38	2.66
	MARS*	85.14	9.31	100.00	0.00	93.96

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\hat{D}} \left[\mathcal{L}_{\text{CE}}\left(F(x;\theta), y\right)\right] + \lambda \sum_{l\in L} \sum_{k\in n_l} BE_k^{(l)}(F(.;\theta)), \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ denotes the cross-entropy loss function, $\hat{D}$ consists of both clean and backdoor samples, and λ represents the regularization coefficient. As shown in Table 3, when λ is set to 0.01 or lower, MARS can effectively defend against adaptive attacks, achieving a CAD of over 93%. We hypothesize that this is due to the small regularization coefficient, which provides limited constraint on the backdoor energy of neurons. However, when λ is further increased to 0.05 or higher, the backdoor energy of malicious models becomes sufficiently constrained, even falling below that of benign models. This causes MARS to misclassify all benign models as malicious, and vice versa , as indicated by a TPR of 0% and an FPR close to 100%. Nevertheless, these results also suggest that even with constrained backdoor energy, there remain significant differences between the CBE distributions of backdoor and benign models. Therefore, we modify MARS's cluster selection strategy from choosing the cluster with the smaller center norm to a majority-based selection, which we refer to as MARS*. We observe that regardless of the λ value, MARS* consistently and effectively defends against adaptive attacks.

Comparison with BackdoorIndicator. The most recently proposed defense BackdoorIndicator [21] identifies that subsequent backdoor injections significantly slow down the ASR decline of previously implanted backdoors. Building on this observation, it employs an indicator task that uses OOD samples to detect and remove backdoored models. As shown in Table 4, BackdoorIndicator effectively detects most backdoor models under the MRA attack, maintaining a low ASR. However, when confronted with the CerP attack, it can only detect a limited number of backdoor models, resulting in an ASR close to 72%, indicating that BackdoorIndicator fails

Table 4: Comparison of MARS and BackdoorIndicator. G and C100 refer to the use of GTSRB and CIFAR-100 as the indicator datasets of BackdoorIndicator, respectively.

Attack	Defense	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
MRA	**Indicator (G)**	85.28	9.32	97.50	0.00	93.37
	Indicator (C100)	85.43	10.29	90.00	0.00	91.29
	MARS	85.16	9.40	100.00	0.00	93.94
CerP	**Indicator (G)**	85.22	71.94	37.50	0.63	62.54
	Indicator (C100)	84.89	71.98	47.50	0.63	64.95
	MARS	85.37	10.03	100.00	0.00	93.84
3DFed	**Indicator (G)**	83.77	96.65	0.00	53.75	33.34
	Indicator (C100)	84.39	97.93	0.00	17.50	42.24
	MARS	85.07	9.86	100.00	0.00	93.80

to provide sufficient protection in this case. Against the 3DFed attack, similar to other evaluated SOTA defenses, BackdoorIndicator completely breaks down, achieving less than half the CAD of MARS. We hypothesize that this is because BackdoorIndicator is a heuristic algorithm that validates its intuition based solely on unconstrained backdoor training. As a result, it performs well against attacks like MRA, which rely solely on data poisoning, a finding supported by both the original paper and our experimental results. However, CerP and 3DFed introduce various constraints during the backdoor model training process, making the attacks more subtle and potent. These constraints likely lead to failures in BackdoorIndicator's underlying intuition, rendering it less effective against these more sophisticated attacks.

Impact of attacker ratio on MARS. Previously, we demonstrated that MARS outperforms existing defenses in terms of resilience to various attacks with a 20% attacker proportion (*i.e.*, the effectiveness goal). To further investigate MARS's robustness, it is important to explore how it performs across a broader range of attacker proportions. Specifically, we aim to examine if MARS mistakenly excludes benign models when there are no attackers (*i.e.*, the fidelity goal) and whether it can still provide effective defense when the attacker proportion exceeds 50% (*i.e.*, the practicability goal). Table 5 presents the evaluation metrics of MARS as the attacker proportion increases from 0% to 95%. Remarkably, MARS consistently identifies all attackers with a TPR of 100%, while ensuring no benign models are misclassified as malicious (FPR of 0%) across all settings. We attribute MARS's outstanding performance in extreme scenarios (*e.g.*, 95% attacker presence) to its carefully designed cluster selection strategy, which utilizes the cluster center norm to identify the trusted cluster and decides whether to discard a cluster based on inter-cluster distance.

Table 5: Impact of attacker ratio on MARS.

Atk. Ratio	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
0	85.26	9.34	100.00	0.00	93.98
10	85.21	9.42	100.00	0.00	93.95
20	85.07	9.86	100.00	0.00	93.80
30	85.13	9.47	100.00	0.00	93.92
50	84.95	9.59	100.00	0.00	93.84
70	84.83	10.54	100.00	0.00	93.57
95	82.99	11.42	100.00	0.00	92.89

Other results. Due to space constraints, we have included additional results in the appendix. Specifically, Appendix G evaluates the impact of data distribution on the performance of existing defenses, Appendix H assesses MARS's sensitivity to hyperparameters, and Appendix I examines MARS's effectiveness on larger datasets such as ImageNet [9]. Appendix J evaluates the computational and communication overhead of MARS. Appendix K assesses the performance of MARS on NLP tasks, while Appendix L and Appendix M examines its defense capabilities against additional Backdoor attacks and Byzantine attacks respectively. Appendix N validates the effectiveness of MARS on ViT.

6 Conclusion and Limitation

We propose MARS, a malignity-aware backdoor defense. Unlike existing defenses that rely on loosely backdoor-coupled empirical statistical metrics, MARS directly focuses on the core nature of backdoor attacks by detecting malignity through the backdoor energies of neurons. We further amplify this malignity by extracting the most prominent backdoor energies. A novel Wasserstein-based clustering method is then introduced to accurately detect backdoored models. Comprehensive comparisons across 3 datasets, 3 SOTA attacks, and 8 SOTA defenses demonstrate the superiority of MARS. Moreover, we validate the robustness of MARS against adaptive attack, further showcasing its effectiveness in backdoor defense. However, MARS is specifically designed for backdoor attacks; it is not well-suited to detect other types of threats that do not directly impact model performance. For instance, it is not designed to handle free-rider attacks [11], where clients may behave lazily without degrading overall performance. Similarly, its defense mechanism does not extend to privacy-stealing attacks, such as gradient inversion [14], which aim to reconstruct training data from shared updates rather than corrupting the model's integrity.

Acknowledgements

Shengshan'work is supported by the National Natural Science Foundation of China under Grant 62372196.

References

[1] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn M. Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST'22)*, 13(4):54:1–54:23, 2022.

[2] Priya Arora, Vikas Khullar, Isha Kansal, Rajeev Kumar, and Renu Popli. Privacy-preserving federated learning system (f-ppls) for military focused area classification. *Multimedia Tools and Applications*, pages 1–27, 2024.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS'20)*, pages 2938–2948, 2020.

[4] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Processings of the 33rd Annual Conference on Neural Information Processing Systems (NeurIPS'19)*, pages 8632–8642, 2019.

[5] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS'17)*, pages 119–129, 2017.

[6] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *Proceedings of the 28th Annual Network and Distributed System Security Symposium (NDSS'21)*, 2021.

[7] Pushpita Chatterjee, Debashis Das, and Danda B. Rawat. Federated learning empowered recommendation model for financial consumer services. *IEEE Transactions on Consumer Electronics (TCE'24)*, 70(1):2508–2516, 2024.

[8] Yanbo Dai and Songze Li. Chameleon: Adapting to peer images for planting durable backdoors in federated learning. In *International Conference on Machine Learning (ICML'23)*, pages 6712–6725, 2023.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'09)*, pages 248–255, 2009.

[10] Christian Fachola, Agustín Tornaría, Paola Bermolen, Germán Capdehourat, Lorena Etcheverry, and María Inés Fariello. Federated learning for data analytics in education. *Data*, 8(2):43, 2023.

[11] Yann Fraboni, Richard Vidal, and Marco Lorenzi. Free-rider attacks on model aggregation in federated learning. In *The 24th International Conference on Artificial Intelligence and Statistics (AISTATS'21)*, pages 1846–1854, 2021.

[12] Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID'20)*, pages 301–316, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'16)*, pages 770–778, 2016.

[14] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. In *Processings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS'21)*, pages 7232–7241, 2021.

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[17] Haoyang Li, Qingqing Ye, Haibo Hu, Jin Li, Leixia Wang, Chengfang Fang, and Jie Shi. 3dfed: Adaptive and extensible framework for covert backdoor attack in federated learning. In *Proceedings of the 44th IEEE Symposium on Security and Privacy (SP'23)*, pages 1893–1907, 2023.

[18] Minghui Li, Wei Wan, Jianrong Lu, Shengshan Hu, Junyu Shi, Leo Yu Zhang, Man Zhou, and Yifeng Zheng. Shielding federated learning: Mitigating byzantine attacks with less constraints. In *Proceedings of 18th International Conference on Mobility, Sensing and Networking (MSN'22)*, pages 178–185, 2022.

[19] Minghui Li, Wei Wan, Yuxuan Ning, Shengshan Hu, Lulu Xue, Leo Yu Zhang, and Yichen Wang. Darkfed: A data-free backdoor attack in federated learning. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence (IJCAI'24)*, pages 4443–4451, 2024.

[20] Minghui Li, Hangtao Zhang, Yanjun Zhang, Li Zeng, Chao Chen, Qiyun Shao, Wei Wan, Shengshan Hu, and Leo Yu Zhang. Fine-grained poisoning framework against federated learning. *IEEE Transactions on Dependable and Secure Computing (TDSC'25)*, 2025.

[21] Songze Li and Yanbo Dai. Backdoorindicator: Leveraging OOD data for proactive backdoor detection in federated learning. In *Proceedings of the 33rd USENIX Security Symposium, (USENIX Security'24)*, 2024.

[22] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys'20)*, 2020.

[23] Jianrong Lu, Lulu Xue, Wei Wan, Minghui Li, Leo Yu Zhang, and Shengqing Hu. Preserving privacy of input features across all stages of collaborative learning. In *Proceedings of the 21st IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA'23)*, pages 191–198, 2023.

[24] Jianrong Lu, Shengshan Hu, Wei Wan, Minghui Li, Leo Yu Zhang, Lulu Xue, and Hai Jin. Depriving the survival space of adversaries against poisoned gradients in federated learning. *IEEE Transactions on Information Forensics and Security (TIFS'24)*, 19:5405–5418, 2024.

[25] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI'23)*, pages 9020–9028, 2023.

[26] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS'17)*, pages 1273–1282, 2017.

[27] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: taming backdoors in federated learning. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security'22)*, pages 1415–1432, 2022.

[28] Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

[29] Ashwinee Panda, Saeed Mahloujifar, Arjun Nitin Bhagoji, Supriyo Chakraborty, and Prateek Mittal. Sparsefed: Mitigating model poisoning attacks in federated learning with sparsification. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS'22)*, pages 7587–7624, 2022.

[30] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. In *Proceedings of the 29th Annual Network and Distributed System Security Symposium (NDSS'22)*, 2022.

[31] Junyu Shi, Wei Wan, Shengshan Hu, Jianrong Lu, and Leo Yu Zhang. Challenges and approaches for mitigating byzantine attacks in federated learning. In *Proceedings of International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom'22)*, pages 139–146. IEEE, 2022.

[32] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *Proceedings of the 25th European Symposium on Research in Computer Security (ESORICS'20)*, pages 480–501, 2020.

[33] Wei Wan, Jianrong Lu, Shengshan Hu, Leo Yu Zhang, and Xiaobing Pei. Shielding federated learning: A new attack approach and its defense. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'21)*, pages 1–7, 2021.

[34] Wei Wan, Shengshan Hu, Jianrong Lu, Leo Yu Zhang, Hai Jin, and Yuanyuan He. Shielding federated learning: Robust aggregation with adaptive client selection. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI'22)*, pages 753–760, 2022.

[35] Wei Wan, Shengshan Hu, Minghui Li, Jianrong Lu, Longling Zhang, Leo Yu Zhang, and Hai Jin. A four-pronged defense against byzantine attacks in federated learning. In *Proceedings of the 31st ACM International Conference on Multimedia (MM'23)*, pages 7394–7402, 2023.

[36] Wei Wan, Yuxuan Ning, Shengshan Hu, Lulu Xue, Minghui Li, Leo Yu Zhang, and Hai Jin. Misa: Unveiling the vulnerabilities in split federated learning. In *proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'24)*, pages 6435–6439, 2024.

[37] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP'19)*, pages 707–723, 2019.

[38] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS'20)*, 2020.

[39] Xianlong Wang, Hewen Pan, Hangtao Zhang, Minghui Li, Shengshan Hu, Ziqi Zhou, Lulu Xue, Peijin Guo, Yichen Wang, Wei Wan, Aishan Liu, and Leo Yu Zhang. Trojanrobot: Backdoor attacks against robotic manipulation in the physical world. *arXiv preprint arXiv: 2411.11683*, 2024.

[40] Yongkang Wang, Dihua Zhai, Yufeng Zhan, and Yuanqing Xia. Rflbat: A robust federated learning algorithm against backdoor attack. *arXiv preprint arXiv:2201.03772*, 2022.

[41] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: distributed backdoor attacks against federated learning. In *Proceedings of the 8th International Conference on Learning Representations (ICLR'20)*, 2020.

[42] Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3FL: adversarially adaptive backdoor attacks to federated learning. In *Annual Conference on Neural Information Processing Systems (NeurIPS'23)*, 2023.

[43] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'22)*, pages 2545–2555, 2022.

[44] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *Proceedings of the 17th European Conference on Computer Vision (ECCV'22)*, pages 175–191, 2022.

[45] Ziqi Zhou, Menghao Deng, Yufei Song, Hangtao Zhang, Wei Wan, Shengshan Hu, Minghui Li, Leo Yu Zhang, and Dezhong Yao. Darkhash: A data-free backdoor attack against deep hashing. *IEEE Transactions on Information Forensics and Security (TIFS'25)*, 2025.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We included the paper's contributions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Sec. 6

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Please refer to Sec. 4.3.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We included the details of our algorithm in Appendix C and experimental setup in Sec. 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is included in supplementary.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Sec. 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We used the same random seed for all defense algorithms to enable a fair comparison of their performance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Sec. 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We acknowledge the Code of Ethics and obey them in our paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited the datasets and models used in our paper. All datasets and models used in our paper are publicly available.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: : The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: the core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Proof of Theorem 1

Theorem 1 (Upper Bound of Backdoor Energy). *Suppose an L-layer neural network F and its every sub-network $f^{(l)}, l \in [1, L]$, are Lipschitz smooth. Then, the backdoor energy of the k^{th} neuron in the l^{th} layer can be upper bounded by:*

$$BE_k^{(l)}(F) \leq \|f_k^{(l)}\|_{\text{Lip}} \left(\prod_{i=1}^{l-1} \|f^{(i)}\|_{\text{Lip}} \right) \mathbb{E}_{x \sim \mathcal{X}} \left[\|x - \delta(x)\|_2 \right],$$

where $\|.\|_{Lip}$ represents the Lipschitz constant of a function.

Proof:

Since each sub-network $f^{(i)}, i \in [1, L]$ is Lipschitz smooth, for all x, y, we have:

$$\|f^{(i)}(x) - f^{(i)}(y)\|_2 \leq \|f^{(i)}\|_{\text{Lip}} \|x - y\|_2$$

Consider the difference in the k-th neuron of the l-th layer between clean and backdoor inputs:

$$\begin{aligned}
\|F_k^{(l)}(x) - F_k^{(l)}(\delta(x))\|_2 &= \|f_k^{(l)} \circ f^{(l-1)} \circ \cdots \circ f^{(1)}(x) - f_k^{(l)} \circ f^{(l-1)} \circ \cdots \circ f^{(1)}(\delta(x))\|_2 \\
&\leq \|f_k^{(l)}\|_{\text{Lip}} \|f^{(l-1)} \circ \cdots \circ f^{(1)}(x) - f^{(l-1)} \circ \cdots \circ f^{(1)}(\delta(x))\|_2 \\
&\leq \|f_k^{(l)}\|_{\text{Lip}} \|f^{(l-1)}\|_{\text{Lip}} \|f^{(l-2)} \circ \cdots \circ f^{(1)}(x) - f^{(l-2)} \circ \cdots \circ f^{(1)}(\delta(x))\|_2 \\
&\leq \cdots \\
&\leq \|f_k^{(l)}\|_{\text{Lip}} \left(\prod_{i=1}^{l-1} \|f^{(i)}\|_{\text{Lip}} \right) \|x - \delta(x)\|_2
\end{aligned}$$

In the proof above, we apply the Lipschitz smooth assumption layer by layer from the outermost to the innermost layers of the network. When considering an outer layer, all remaining inner sub-networks are treated as a single entity.

Taking the expectation over $x \sim \mathcal{X}$:

$$BE_k^{(l)}(F) = \mathbb{E}_{x \sim \mathcal{X}} \left[\|F_k^{(l)}(x) - F_k^{(l)}(\delta(x))\|_2 \right] \leq \|f_k^{(l)}\|_{\text{Lip}} \left(\prod_{i=1}^{l-1} \|f^{(i)}\|_{\text{Lip}} \right) \mathbb{E}_{x \sim \mathcal{X}} \left[\|x - \delta(x)\|_2 \right]$$

Thus, we conclude the proof.

□

B Distinctions between MARS and CLP

Although CLP [44] also employs a Lipschitz constant to identify suspicious neurons, MARS diverges from CLP in its fundamental approach. First, we derive a rigorous theoretical upper bound on the backdoor energy (BE), furnishing a solid mathematical guarantee, whereas CLP merely uncovers—through empirical observations—a positive correlation between the UCLC (Upper bound of Channel Lipschitz Constant) metric and TAC (Trigger-Activated Change). Second, CLP's use of a Lipschitz-based estimate for BE to prune high-energy neurons is inherently imprecise: overly aggressive pruning can degrade clean accuracy, while insufficient pruning allows a high attack success rate, as evidenced by the results in Sec. 5.2. In contrast, MARS treats BE purely as a feature-extraction tool for downstream clustering, enabling more reliable and accurate detection of backdoored models.

C Calculation of Lipschitz Constant

Assume that a certain subnetwork f (for convenience, we omit the layer index l) is linear, i.e., $f(x) = Wx + b$. According to the definition of the Lipschitz constant, $\|f\|_{\text{Lip}} = \underset{\Delta x \neq 0}{max} \frac{\|f(x+\Delta x) - f(x)\|_2}{\|\Delta x\|_2} = \underset{\Delta x \neq 0}{max} \frac{\|W \cdot \Delta x\|_2}{\|\Delta x\|_2}$. The rightmost part of this equation is precisely the spectral norm of the matrix W,

which can be computed using Singular Value Decomposition (SVD). Specifically, we decompose matrix W into the product of three matrices: one orthogonal matrix, one diagonal matrix, and another orthogonal matrix. The largest element in the diagonal matrix is the spectral norm of W, i.e., $\|f\|_{\text{Lip}}$.

In PyTorch, the Lipschitz constant can be easily computed using torch.svd(weight)[1].max(). For fully-connected layers, we can directly apply the above method. For convolutional layers, we approximate them as linear, and then reshape the weight tensor into a matrix form. The spectral norm of the reshaped matrix is used as an approximation to the original spectral norm. For batch normalization (BN) layers (assuming the BN transformation is $y = \frac{x-\mu}{\sigma} \cdot \gamma + \beta$), we use $\|\frac{\gamma}{\sigma}\|$ to estimate the Lipschitz constant, as it reflects the maximum possible scaling of the input variation after passing through the BN layer, which aligns with the core purpose of the Lipschitz constant. Additionally, to enhance the reproducibility of MARS, we will open-source the code as soon as the paper is accepted.

D Algorithm Description

Algorithm 1 provides a detailed description of MARS. The central server first calculates the backdoor energy (BE) for all neurons in each local model (Lines 1-8), then extracts the most prominent BE values from each layer to form the concentrated backdoor energy (CBE), which is stored in a set A (Line 9). Using the Wasserstein-based clustering algorithm, the server clusters all local models' CBEs into two groups based on the CBEs in set A, storing the client indices of each cluster in S_1 and S_2, respectively (Line 11). The centers A_1 and A_2 of the two clusters are computed (Lines 12-13). If the Wasserstein distance between A_1 and A_2 is within an acceptable threshold ϵ, it indicates that the distributions of the two clusters are similar, and thus all local models are considered benign (Lines 14-15). Otherwise, the local models corresponding to the cluster with the smaller norm of the cluster center are used for aggregation (Lines 16-23).

E Evaluated Attacks and Defenses

E.1 Attacks

MRA [3]. MRA (Model Replacement Attack) is the first backdoor attack specifically designed for FL. Its core idea is to amplify backdoor updates in proportion, allowing a small number of malicious updates to dominate the global model. MRA is widely used for assessing the robustness of backdoor defenses in FL.

CerP [25]. CerP (Cerberus Poisoning) is an advanced backdoor attack algorithm for FL that has emerged in recent years. It simultaneously tunes the backdoor trigger while controlling the changes in the poisoned model for each malicious participant, enabling a stealthy yet effective backdoor attack against a wide range of FL defense mechanisms. By fine-tuning the trigger, CerP increases the compatibility between the backdoor model and the trigger, minimizing significant updates to the model parameters. Additionally, controlling the changes in the model reduces the disparity between the backdoor and benign models, making it more challenging for defenders to identify the backdoor models.

3DFed [17]. 3DFed is an adaptive and extensible framework designed for launching covert backdoor attacks in FL environments, particularly in a black-box setting. It addresses the challenges posed by existing backdoor attacks, which often require extensive information about the victim FL system and typically optimize for a single objective, rendering them less effective against sophisticated defense mechanisms. The core of 3DFed lies in its three evasion modules that effectively camouflage backdoor models: backdoor training with constrained loss, noise mask, and decoy model. These components work synergistically to implant indicators into the backdoor model, allowing 3DFed to capture attack feedback from the global model during the previous training epoch. This feedback enables dynamic adjustment of hyper-parameters within the evasion modules, enhancing the stealth and efficacy of the attacks. To the best of our knowledge, MARS is the first defense to conduct a comprehensive evaluation of robustness against 3DFed.

Algorithm 1 MARS

Require: Set of selected clients in the current round: S;
Set of corresponding local models: $\{F(.;\theta_s), s \in S\}$; `# We omit round index` t `for simplicity`
Top factor: κ;
Inter-cluster threshold: ϵ.
Ensure: Aggregated global model: $F(.;\theta^G)$.

Initialize set $A \leftarrow \{\}$ `# Set` A `is used to preserve the CBE of local models`
for $s \in S$ **do**
  $\theta \leftarrow \theta_s$
  for $l \in L$ **do**
    for $k \in n_l$ **do**
      $BE_k^{(l)}(F(.;\theta)) = \|f_k^{(l)}\|_{Lip}$ `# Calculate BE of each neuron via eq. 3`
    end for
  end for
  $A[s] \leftarrow \bigcup_{l=1}^{L} \text{TopK}_{\kappa\%}\left(\{BE_i^{(l)}(F(.;\theta))\}_{i=1}^{n_l}\right)$ `# Calculate CBE for each client via eq. 4`
end for
$S_1, S_2 \leftarrow$ **K-WMeans**(A) `# Divide client index into two clusters` S_1 `and` S_2
$A_1 \leftarrow$ **Mean**$(\{A[s], s \in S_1\})$
$A_2 \leftarrow$ **Mean**$(\{A[s], s \in S_2\})$
if $Wass(A_1, A_2) < \epsilon$ **then**
  $S_{\text{final}} \leftarrow S$ `# All the global models are used for aggregation`
else
  if $\|A_1\|_1 < \|A_2\|_1$ **then**
    $S_{\text{final}} \leftarrow S_1$ `# Preserve the cluster with lower norm of central CBE`
  else
    $S_{\text{final}} \leftarrow S_2$
  end if
end if
$\theta^G \leftarrow \frac{1}{|S_{\text{final}}|} \sum_{s \in S_{\text{final}}} \theta_s$ `# Aggregate all the credible local model weights`
return $F(.;\theta^G)$

E.2 Defenses

FedAvg [26]. FedAvg is the first aggregation algorithm for FL, which constructs a high-performance global model by aggregating all local models through weighted averaging. Due to its effective knowledge aggregation capabilities, FedAvg is widely utilized in real-world industrial applications, such as Google's GBoard. Consequently, existing works usually evaluate the resistance of FedAvg to backdoor attacks, making it a critical baseline for comparison.

Multi-Krum [5]. Multi-Krum is a defense algorithm based on out-of-distribution (OOD) detection. It estimates whether a local model deviates from the overall distribution by calculating the sum of distances between that model and its nearest $n - f - 2$ neighbor models (n and f represent the number of participants and the number of attackers, respectively). Subsequently, it excludes the models that are furthest from the overall distribution from the aggregation queue.

RFLBAT [40]. RFLBAT is a cutting-edge defense mechanism designed to counteract backdoor attacks in FL systems. Unlike existing algorithms that often impose constraints on the number of malicious attackers or assume independent and identically distributed (IID) data, RFLBAT operates effectively under realistic conditions where the number of attackers is unknown and the data distribution is typically non-IID. RFLBAT leverages principal component analysis (PCA) to identify and extract essential features from the model updates, followed by a K-means clustering algorithm to group similar updates. This dual approach enables RFLBAT to effectively filter out malicious updates without requiring additional auxiliary information beyond the learning process itself.

FLAME [27]. FLAME is a defense framework aimed at countering backdoor attacks in FL. The key implementation steps of FLAME are as follows: *Noise Estimation.* FLAME estimates the

optimal amount of noise to inject, ensuring effective elimination of backdoors while preserving model performance. *Model Clustering.* The framework utilizes a clustering approach to group similar models, which helps identify and isolate potentially malicious updates. *Weight Clipping.* FLAME applies weight clipping to the clustered models, mitigating the influence of adversarial updates and maintaining the integrity of the aggregated model. Through these steps, FLAME effectively defends against backdoor attacks with minimal impact on the performance of benign updates.

FoolsGold [12]. FoolsGold is a consistency detection-based defense that identifies poisoning updates based on the diversity of client updates in the distributed learning process. Specifically, Updates with excessively high pairwise cosine similarity are assigned lower aggregation weights. Unlike prior work, FoolsGold does not bound the expected number of attackers, requires no auxiliary information outside of the learning process, and makes fewer assumptions about clients and their data.

FLDetector [43]. FLDetector is a defense mechanism designed to address the challenge of model poisoning attacks in FL, particularly when there is a large number of malicious clients. The core insight behind FLDetector is that model poisoning attacks lead to inconsistent updates from malicious clients across multiple iterations. To identify these inconsistencies, FLDetector predicts each client's model update in subsequent iterations based on its historical updates and flags a client as malicious if its updates deviate from the predicted values across several iterations. This approach allows FLDetector to accurately detect and remove malicious clients, ensuring that existing robust FL methods can continue to function effectively even under strong attack scenarios.

DeepSight [30]. DeepSight is a model filtering approach designed to mitigate backdoor attacks in FL without removing benign models from clients with diverse data distributions. Unlike existing defenses that simply exclude deviating models, DeepSight introduces three novel techniques to better characterize the data distribution behind model updates and measure subtle differences in the internal structure and outputs of neural networks (NNs). These techniques allow DeepSight to detect suspicious model updates effectively. Additionally, it employs a clustering scheme to group models and identify clusters that contain poisoned updates with high attack impact. By combining these insights, DeepSight can eliminate harmful model clusters, while also mitigating any residual backdoor effects using weight clipping defenses.

FedCLP [44]. CLP (Lipschitzness based Pruning) is a novel approach designed to detect and remove backdoor channels in deep neural networks (DNNs) without requiring any data. It introduces the concept of the Channel Lipschitz Constant (CLC), which measures the Lipschitz constant of the mapping from input images to the output of each channel. By analyzing the correlation between an upper bound of the CLC (UCLC) and the activation changes caused by a backdoor trigger, CLP identifies potential backdoor channels. Since UCLC can be directly computed from the network's weight matrices, CLP operates in a completely data-free manner. Once these infected channels are detected, CLP prunes them to repair the model. This method is fast, simple, and robust, making it an efficient solution for backdoor defense with minimal dependency on the choice of the pruning threshold. We adapt CLP to the FL setting and name it FedCLP. Specifically, we prune each local model using CLP to remove backdoor-related information before aggregating them with FedAvg.

BackdoorIndicator [21]. BackdoorIndicator is a proactive backdoor detection mechanism specifically designed for FL systems. This mechanism operates on the insight that deploying subsequent backdoors with the same target label can enhance the accuracy of existing backdoors. BackdoorIndicator enables the server to inject indicator tasks into the global model using out-of-distribution (OOD) data. Since any backdoor samples are inherently OOD concerning benign samples, the server, unaware of the specific backdoor types or target labels, can effectively detect backdoor presence in uploaded models by evaluating the performance of these indicator tasks. Through comprehensive empirical evaluations, BackdoorIndicator demonstrates consistently superior performance and practicality compared to existing baseline defenses across various system configurations and adversarial scenarios.

F Evaluation Metrics

We evaluate the performance of a defense using four metrics: ACC, ASR, TPR, and FPR, each providing distinct perspectives on the effectiveness of the defense. Additionally, based on these metrics, we introduce a novel metric called CAD, which offers a comprehensive view of the overall effectiveness of the defense.

ACC. ACC (Model Accuracy) is calculated as the proportion of correctly identified clean samples to the total number of clean samples. In federated learning, maintaining high accuracy is crucial, as it reflects the model's overall performance in making correct predictions across all clients.

ASR. ASR (Attack Success Rate) measures the proportion of samples with triggers that are classified as the target label. A lower ASR indicates that the defense mechanism is effective in identifying and mitigating backdoor attacks. In federated learning scenarios, minimizing ASR is essential to ensure the system remains resilient against adversarial manipulation. It is important to note that in our experiments, we do not exclude samples corresponding to the target label. As a result, even for a clean model, the ASR does not approach 0 but rather tends toward $1/c$ (c represents the total number of classes).

TPR. TPR (True Positive Rate) measures the proportion of backdoor models that are correctly identified by the defense algorithm as malicious. High TPR is indicative of the defense algorithm's effectiveness in accurately detecting backdoor models. A robust defense mechanism should achieve high TPR to minimize the risk of allowing backdoor attacks to compromise the integrity of the model. This is critical for maintaining trust and reliability in federated learning environments.

FPR. FPR (False Positive Rate) measures the proportion of legitimate models that are incorrectly classified by the defense algorithm as backdoored. A low FPR is crucial as it indicates that the defense algorithm does not mistakenly flag benign models as backdoored. In the context of federated learning, minimizing FPR is essential to prevent unnecessary disruptions to legitimate model updates and to maintain the overall functionality of the system.

CAD. CAD (Comprohensive Abilisty of Defense) is a composite metric that integrates the four aforementioned indicators to provide an overall assessment of a defense algorithm's performance. It is calculated as follows:

$$CAD = \frac{ACC + (1 - ASR) + TPR + (1 - FPR)}{4} \times 100\%. \tag{7}$$

This formulation captures a balanced view of accuracy, attack resistance, true positive detection, and false positive minimization.

It is important to note that FoolsGold does not directly discard local models but assigns lower aggregation weights to suspected models. When calculating its TPR and FPR, we consider local models with an aggregation weight greater than 0.5 as selected by FoolsGold, otherwise, the model is deemed rejected. Additionally, FedCLP does not distinguish between benign and backdoor models, instead pruning all local models before aggregation. Therefore, TPR and FPR cannot be calculated for FedCLP, and we denote the values of these metrics as "-". For the CAD calculation of FedCLP, we only consider ACC and ASR, *i.e.*, $CAD = \frac{ACC+(1-ASR)}{2} \times 100\%$.

G Impact of data distribution

The previous experiments are conducted using a Dirichlet sampling parameter of $\alpha = 0.9$, which is the default setting recommended by 3DFed. To assess the impact of a broader range of data distributions on the performance of existing defenses, we follow BackdoorIndicator by considering three non-IID data distributions with α values of 0.2, 0.5, and 0.9. Notably, a smaller α indicates a higher degree of data heterogeneity. Additionally, we examine an IID data distribution ($\alpha = 10$), a scenario often overlooked by existing defenses. Figure 3 illustrates the data distribution of each client under different values of alpha. As shown in Table 6, overall, the performance of existing defenses gradually deteriorates as data heterogeneity increases. For instance, FLAME effectively counters 3DFed attacks with a CAD of 86.09% at $\alpha = 10$, but it completely fails in non-IID scenarios, with a CAD of only around 30%. While FedCLP can mitigate backdoor attacks, it also leads to varying degrees of ACC reduction, with more significant drops in non-IID settings. Interestingly, we observe a counterintuitive phenomenon where FLDetector performs worse in IID scenarios; we speculate that this is because 3DFed makes fewer modifications to the backdoor models in IID settings, making the predicted models and backdoor models more similar, which causes FLDetector to mistakenly classify benign models as backdoor models. MARS consistently performs excellently across all data distributions, with a CAD always above 93%.

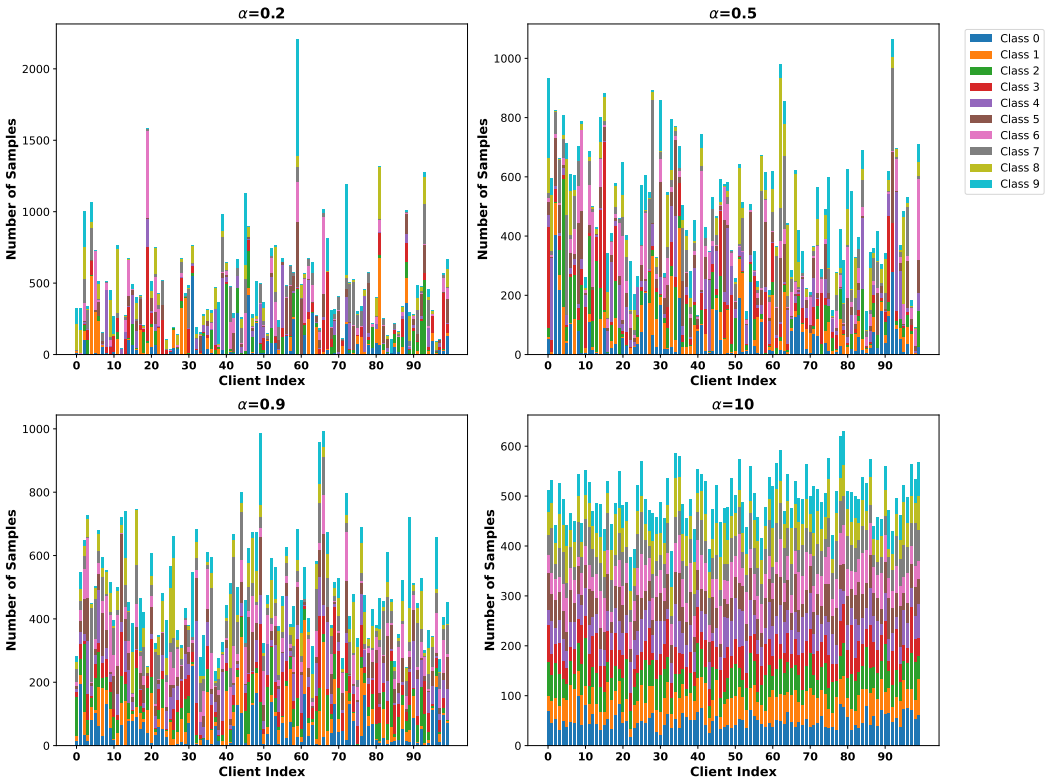


Figure 3: Dirichlet sampling with different α.

Table 6: Impact of data distribution on the performance of exitsting defenses under 3DFed attack on CIFAR-10.

α	Metric	FedAvg	MultiKrum	RFLBAT	FLAME	FoolsGold	FLDetector	DeepSight	FedCLP	MARS
0.2	ACC	83.38	82.47	82.82	80.08	82.66	82.04	76.65	60.87	83.26
	ASR	97.37	99.44	95.75	97.22	93.97	93.48	98.32	4.72	9.38
	TPR	0.00	17.50	0.00	0.00	30.00	0.00	0.00	-	100.00
	FPR	0.00	20.63	0.00	56.25	60.63	0.00	43.75	-	0.00
	CAD	46.50	44.98	46.77	31.65	39.51	47.14	33.65	78.07	93.47
0.5	ACC	84.24	83.88	83.58	83.41	83.60	84.89	83.89	73.28	84.66
	ASR	98.39	96.40	97.49	96.72	98.28	91.79	90.11	10.28	9.90
	TPR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	100.00
	FPR	0.00	25.00	0.00	56.25	68.75	0.00	6.25	-	0.00
	CAD	46.46	40.62	46.52	32.61	29.14	48.28	46.88	81.50	93.69
0.9	ACC	84.37	84.07	84.30	83.06	84.11	84.24	84.80	69.25	85.07
	ASR	96.76	97.27	92.02	97.50	96.29	95.20	98.85	7.55	9.86
	TPR	0.00	0.00	0.00	2.50	0.00	0.00	0.00	-	100.00
	FPR	0.00	25.00	5.00	55.63	0.25	35.00	6.25	-	0.00
	CAD	46.90	40.45	46.82	33.11	46.89	38.51	44.93	80.85	93.80
10	ACC	84.60	84.51	85.19	85.28	84.64	83.82	84.67	76.20	85.30
	ASR	95.68	98.54	76.18	9.67	96.36	99.13	70.08	8.75	9.49
	TPR	0.00	5.00	20.00	100.00	0.00	0.00	0.00	-	100.00
	FPR	0.00	23.75	11.25	31.25	0.00	100.00	6.25	-	0.00
	CAD	47.23	41.81	54.44	86.09	47.07	21.17	52.09	83.73	93.95

H Sensitivity to hyperparameters

H.1 Impact of distance metric

In Section 4.5, we illustrate with a toy example that Wasserstein distance is more suitable for MARS compared to traditional Euclidean and cosine distances. To further substantiate our claim, we replace MARS's distance metric with Euclidean distance and cosine distance, keeping all other components constant. As shown in Table 7, both Euclidean and cosine distances fail to accurately detect backdoor updates, resulting in a CAD of only around 44%. In contrast, when using Wasserstein distance, MARS achieves optimal performance with a CAD close to 94%. This supports our hypothesis that Wasserstein distance, which is insensitive to the order of elements, is more effective for detecting backdoor models in our scenario.

Table 7: Impact of distance metric on MARS under CerP attack on CIFAR-10.

Dist.	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
Euc.	83.93	88.15	35.00	51.25	44.88
Cos.	84.29	82.05	32.50	58.13	44.15
Wass.	85.37	10.03	100.00	0.00	93.84

H.2 Sensitivity to ϵ

In Section 4.5, to avoid blindly removing a cluster in non-adversarial scenarios, which could degrade model accuracy, we propose using inter-cluster distance to decide whether to retain all clusters, with an acceptable threshold set to ϵ. As shown in Table 8, in the presence of attackers, MARS accurately distinguishes between benign and malicious models as long as ϵ does not exceed 1. In non-adversarial scenarios, when ϵ is no less than 0.03, MARS does not mistakenly classify any benign models as backdoor models. Therefore, setting ϵ between 0.03 and 1 ensures optimal performance for MARS. The wide range of acceptable ϵ values indicates that MARS is not highly sensitive to this parameter, making it easy to select an appropriate ϵ in real-world scenarios.

Table 8: Impact of ϵ on MARS under 3DFed attack on CIFAR-10.

	Metric	0.01	0.02	0.03	0.04	0.05	0.10	0.50	1.00	3.00	5.00
w/ attack	**TPR**	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	63.64	18.18
	FPR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
w/o attack	**FPR**	42.73	8.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

H.3 Sensitivity to κ

Review that in Section 4.4, in order to concentrate backdoor activity and facilitate subsequent detection of backdoor models, we extract the top $\kappa\%$ of BE values from each layer of local models, forming a one-dimensional vector called CBE. As shown in Table 9, when κ is set to 10 or less, MARS achieves a TPR of 100% and an FPR of 0%, indicating that MARS can precisely detect all backdoor models without mistakenly discarding any benign models. However, when κ exceeds 20, MARS begins to miss some backdoor models, and in some cases, even misidentifies a few benign models as backdoor ones. In real-world deployments, setting κ to 10 or below ensures optimal performance (with the default in this paper being 5), which is easily achievable. Therefore, MARS is not highly sensitive to the choice of κ.

Table 9: Impact of κ on MARS under 3DFed attack on CIFAR-10.

Metric	1	2	5	10	20	40	60	80	100
TPR	100.00	100.00	100.00	100.00	94.44	87.10	83.33	91.67	77.42
FPR	0.00	0.00	0.00	0.00	0.35	0.00	0.42	0.52	0.00

I Performance on ImageNet

In the main text, we evaluate the effectiveness of MARS on MNIST, CIFAR-10, and CIFAR-100, following the common practice in existing defenses such as BackdoorIndicator and FLDetector. However, real-world datasets are typically more complex and challenging. Hence, it is essential to assess the performance of MARS on larger, more intricate datasets. We use ImageNet as the benchmark dataset and ReXNet as the network architecture. Regarding attacks, due to the lack of open-source code compatible with ImageNet for 3DFed and CerP, and after several attempts to adapt their parameters to work with ImageNet without success, we focus solely on the MRA attack. On the defense side, we compare MARS with FedAvg in both adversarial and non-adversarial (referred to as the Baseline) settings. As shown in Table 10, with FedAvg, ASR escalates from 0.14% to 98.54% as training progresses, highlighting the significant threat posed by MRA to federated learning systems. However, when MARS is deployed on the central server, ACC remains consistently above 75%, and ASR is reduced to around 0.1%, comparable to the Baseline. This demonstrates that MARS is effective even when applied to large-scale datasets like ImageNet.

Table 10: Comparison of MARS under MRA attack on ImageNet.

Round	Defense	ACC ↑	ASR ↓	TPR ↑	FPR ↓	CAD ↑
1	**FedAvg**	69.54	0.14	0.00	0.00	67.35
	MARS	75.87	0.10	100.00	0.00	93.94
	Baseline	76.25	0.08	-	-	-
10	**FedAvg**	74.64	1.05	0.00	0.00	68.40
	MARS	75.47	0.12	100.00	0.00	93.84
	Baseline	75.85	0.08	-	-	-
20	**FedAvg**	73.81	19.94	0.00	0.00	63.47
	MARS	75.44	0.12	100.00	0.00	93.83
	Baseline	75.89	0.08	-	-	-
30	**FedAvg**	73.91	84.12	0.00	0.00	47.45
	MARS	75.49	0.12	100.00	0.00	93.84
	Baseline	75.59	0.08	-	-	-
40	**FedAvg**	74.19	95.59	0.00	0.00	44.65
	MARS	75.22	0.12	100.00	0.00	93.78
	Baseline	75.34	0.08	-	-	-
50	**FedAvg**	73.73	98.54	0.00	0.00	43.80
	MARS	75.14	0.12	100.00	0.00	93.76
	Baseline	75.26	0.08	-	-	-

J Computational and Communication Overheads of MARS

MARS does not require clients to upload anything other than model parameters, resulting in no additional communication overhead compared to existing defenses such as FedAvg. In terms of computational overhead, the aggregation time required by MARS (including BE computation, CBE formation, Wasserstein-based clustering, and the final aggregation to obtain the new global model) is shorter than that of most existing defenses. Table 11 presents results on the CIFAR-10 dataset with a ResNet-18 model, a total of 100 clients, 20 of whom are attackers, with 20 clients randomly sampled per round. We recorded the average runtime per round for each defense method. As shown, MARS, FedAvg, and FLAME complete aggregation within 7 seconds, while the other six defenses require longer aggregation time, with DeepSight taking as much as 101.69 seconds. The rapid runtime of MARS is achieved through several key tricks. First, we extract the top-$\kappa\%$ of BE values to form CBE, which significantly reduces the time needed for subsequent Wasserstein-based clustering. Second, we estimate BE values only for convolutional and bn layers, ignoring the fully connected layers that

are the most time-consuming. Third, inspired by the work “Rethinking Lipschitzness for Data-free Backdoor Defense" (submited to ICLR 2025), we optimize the computation of the Lipschitz constant using dot product properties.

Table 11: Average runtime per round

Defense	Time per round (s)
FedAvg	2.07
MultiKrum	28.87
RFLBAT	39.19
FLAME	3.87
FoolsGold	7.05
FLDetector	18.91
DeepSight	101.69
FedCLP	38.81
MARS	6.57

K Performance on NLP Task

In this section, we add an evaluation of MARS on the IMDB dataset, using LSTM as the model structure. As shown in Table 12, MARS is also applicable to text data, achieving performance comparable to FedAvg in the non-adversarial scenario.

Table 12: Performance on IMDB dataset

Defense	ACC	ASR	TPR	FPR	CAD
FedAvg	73.89	100.00	0.00	0.00	43.47
FedAvg (non-adversarial scenario)	74.42	56.87	-	-	-
MARS	74.11	57.91	100.00	0.00	79.05

L Evaluation against More Attacks

In this section, we evaluate four additional attacks. They are Dyn-Attack, A3FL [42], Chameleon [8], sematic backdoor attack [3], and partial layer attack.

L.1 Performance on Dyn-Attack

We introduce a new attack method, named Dyn-Attack. Specifically, each attacker randomly selects one of four strategies: 3DFed, CerP, MRA, or no attack. As shown in Table 13, MARS performs comparably to FedAvg in the non-adversarial scenario.

Table 13: Performance under Dyn-Attack

Defense	ACC	ASR
FedAvg (non-adversarial scenario)	85.26	9.34
MARS	85.10	10.19

L.2 Performance under A3FL

Recently, optimized backdoor attacks have gained widespread attention for enhancing stealth by refining triggers. A3FL stands out as a notable example. Table 14 presents MARS’s defense performance against A3FL, demonstrating its ability to effectively and completely neutralize the attack.

Table 14: Performance under A3FL

Defense	ACC	ASR	TPR	FPR	CAD
FedAvg	83.07	98.53	0.00	0.00	46.14
MARS	85.01	9.92	99.19	0.00	93.57

L.3 Performance under Chameleon

Recently, increasing attention has been given to the persistence of backdoor attacks. Once a backdoor is successfully injected, the global model can maintain a certain attack success rate even if the attacker does not participate in federated learning for multiple rounds. Chameleon is a prominent example of such attacks. To assess MARS's ability to defend against this type of threat, we evaluated its performance under Chameleon attacks. As shown in Table 15, MARS effectively mitigates Chameleon attacks.

Table 15: Performance under Chameleon

Defense	**ACC**	**ASR**	**TPR**	**FPR**	**CAD**
FedAvg	83.61	78.00	0.00	0.00	51.40
FedAvg (non-adversarial scenario)	85.29	10.15	-	-	-
MARS	85.22	11.08	98.89	0.28	93.19

L.4 Performance under semantic backdoor attack

We further evaluate MARS against a semantic backdoor attack on CIFAR-10. In this attack, cars with vertically striped walls in the background are misclassified as birds. As shown in Table 16, MARS effectively mitigates this semantic backdoor attack.

Table 16: Performance under semantic backdoor attack

Defense	**ACC**	**ASR**	**TPR**	**FPR**	**CAD**
FedAvg	85.69	80.00	0.00	51.42	0.00
MARS	85.97	0.00	100.00	96.49	0.00

L.5 Performance under partial layer attack

For efficiency, our implementation of MARS omits detection on the most time-consuming fully-connected layers. This optimization, however, creates a potential vulnerability where an attacker might launch a "partial layer attack." To analyze this threat, we specifically evaluate the following attack strategies.

- FC-only
- 1Conv+FC
- 2Convs+FC
- 3Convs+FC
- 4Convs+FC
- All layers (i.e., full-parameter attacks)

As detailed in Table 17, when an attacker injects a backdoor exclusively into the FC layer, MARS (partial-layers) fails to detect the malicious updates (TPR = 0.00%, FPR = 100.00%), since it ignores the manipulated layers. However, the attack itself is unsuccessful: the ASR drops to just 9.86%, while clean accuracy also suffers. This suggests that injecting a backdoor using only the FC layers fails to achieve both stealth and effectiveness. We hypothesize that this is due to the limited expressive capacity of isolated FC-layer tuning: the convolutional layers generate nearly identical features for a

clean sample and the corresponding triggered sample, making it difficult for the final layer alone to simultaneously satisfy both objectives (i.e., clean accuracy and attack success). When the attacker modifies one or two Conv blocks in addition to the FC layer, the resulting attack is still weak (e.g., ASR = 12.36% and 56.60%, respectively, under FedAvg). Nonetheless, both MARS (partial-layers) and MARS (all-layers) consistently achieve 100% TPR and 0% FPR, demonstrating strong resilience against these more involved but still low-intensity attacks. When more Conv blocks are compromised, and especially when all parameters are manipulated, the attack becomes significantly more effective (e.g., ASR = 99.68%). However, MARS still maintains perfect detection performance, with 100% TPR and 0% FPR in all such cases. This suggests that stronger malicious behavior actually makes detection easier for MARS, further validating its robustness.

Table 17: Performance uner partial layer attack

Attack Strategy	Defense	ACC ↑	ASR ↓	TPR ↑	FPR ↓
FC only	**MARS (partial-layers)**	82.74	9.86	0.00	100.00
	MARS (all-layers)	85.29	9.25	100.00	25.00
	FedAvg	85.33	9.91	0.00	0.00
1Conv+FC	**MARS (partial-layers)**	85.48	9.34	100.00	0.00
	MARS (all-layers)	85.46	9.49	100.00	0.00
	FedAvg	84.51	12.36	0.00	0.00
2Convs+FC	**MARS (partial-layers)**	85.51	9.55	100.00	0.00
	MARS (all-layers)	85.57	9.32	100.00	0.00
	FedAvg	84.32	56.60	0.00	0.00
3Convs+FC	**MARS (partial-layers)**	85.37	9.49	100.00	0.00
	MARS (all-layers)	85.55	9.41	100.00	0.00
	FedAvg	84.94	91.45	0.00	0.00
4Convs+FC	**MARS (partial-layers)**	85.54	9.23	100.00	0.00
	MARS (all-layers)	85.56	9.14	100.00	0.00
	FedAvg	85.32	96.17	0.00	0.00
All layers	**MARS (partial-layers)**	85.16	9.40	100.00	0.00
	MARS (all-layers)	85.28	9.67	100.00	0.00
	FedAvg	78.32	99.68	0.00	0.00

M Evaluation against Byzantine Attacks

Although our paper primarily focuses on backdoor defense in federated learning, we believe MARS also holds promise for resisting Byzantine attacks. As illustrated in MAB-RFL [34], Byzantine defense is essentially an anomaly detection problem in high-dimensional data. One common defense approach is to extract key information from local models to obtain low-dimensional representations, which facilitate the subsequent calculation of anomaly scores or clustering. In MARS, the process of calculating BE/CBE serves a similar purpose by extracting discriminative representations that distinguish benign from malicious models. Intuitively, this suggests that MARS could be effective in mitigating Byzantine failures.

To further validate this intuition, we simulated a CIFAR-10 federated learning scenario with 100 clients, where 20% are attackers, and 20% of clients participate in each round over 100 rounds. We considered two typical Byzantine attacks:

- Label Flipping Attack (LFA) [32]: a data poisoning attack.
- Little Is Enough (LIE) [4]: a model poisoning attack known for its high stealth and destructiveness.

For defense, we compared FedAvg (evaluated in a benign scenario as the baseline), Multi-Krum, and MARS.

In the LFA scenario (see Table 18), MARS achieved a true positive rate (TPR) of 80% and a false positive rate (FPR) of 1.25%, slightly lower in detection performance than Multi-Krum. However, both methods yielded similar global accuracy (ACC), because LFA's relatively low maliciousness means that missing a few malicious updates does not significantly impact ACC.

Table 18: Performance under LFA

Defense	ACC	TPR	FPR
Baseline	63.47	–	–
MARS	60.56	80.00	1.25
FedAvg	53.57	0.00	0.00
Multi-Krum	60.73	100.00	0.00

In contrast, for the more potent and stealthy LIE scenario (see Table 19), Multi-Krum's TPR was 0%—it failed to detect malicious models, and its ACC dropped dramatically (even falling below FedAvg). Meanwhile, MARS achieved 100% TPR and 0% FPR, reliably distinguishing malicious models from benign ones in every round.

Table 19: Performance under LIE

Defense	ACC	TPR	FPR
Baseline	63.47	–	–
MARS	60.87	100.00	0.00
FedAvg	41.26	0.00	0.00
Multi-Krum	34.17	0.00	25.00

It is worth noting that in both attack scenarios, MARS's ACC was approximately 3% lower than the baseline FedAvg. This difference is expected, as the baseline was evaluated under attack-free conditions with a higher proportion of benign clients (approximately 25% more per round), which naturally results in better accuracy and faster convergence.

In summary, while MARS was designed for backdoor defense, its underlying representation-based anomaly detection mechanism suggests that it can also serve as a robust defense against Byzantine adversaries.

Table 20: Performance on ViT

Round	Defense	ACC ↑	ASR ↓	TPR ↑	FPR ↓
1	**FedAvg**	24.79	8.24	0.00	0.00
	MARS	96.98	9.93	100.00	0.00
	Baseline	97.36	10.01	-	-
5	**FedAvg**	94.31	99.59	0.00	0.00
	MARS	97.69	10.00	100.00	0.00
	Baseline	97.91	10.02	-	-
10	**FedAvg**	96.03	99.89	0.00	0.00
	MARS	97.88	9.98	100.00	0.00
	Baseline	98.08	9.98	-	-
15	**FedAvg**	96.62	99.89	0.00	0.00
	MARS	97.97	10.00	100.00	0.00
	Baseline	98.11	9.99	-	-
20	**FedAvg**	96.82	99.85	0.00	0.00
	MARS	97.93	9.99	100.00	0.00
	Baseline	98.08	10.00	-	-

N Performance on ViT

To demonstrate that MARS scales to Vision Transformer architectures, we evaluated it on a pre-trained ViT model using the Hugging Face Transformers library. Specifically, we loaded

```
ViTForImageClassification.from_pretrained('google/vit-base-patch16-224-in21k',
    num_labels=10, ignore_mismatched_sizes=True)
```

and fine-tuned it on CIFAR-10. This ViT contains a very high proportion of linear layers (99.09% of its parameters), so MARS remains fully applicable. As shown in Table 20, MARS on ViT achieves detection performance comparable to Baseline (i.e., FedAvg in attack-free scenario), confirming its effectiveness on large-scale models.