Journal of Information Security and Applications 98 (2026) 104387

Contents lists available at ScienceDirect

Journal of Information Security and Applications

journal homepage: www.elsevier.com/locate/jisa

AMF-CFL: Anomaly model filtering based on clustering in federated learning

Bo Wang [a], Xiaorui Dai [a], Wei Wang [b], Zi Yang [a], Zhaoning Wang [a], Maozhen Zhang [a,*]

[a] School of Information and Communication Engineering, Dalian University of Technology, Dalian, China
[b] Institute of Automation, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Federated learning
Model poisoning attack
Malicious client detection

ABSTRACT

Federated learning (FL) allows multiple participants to collaboratively train a shared model without exposing their local data, thereby mitigating the risk of data leakage. Despite its advantages, FL is vulnerable to attacks by malicious clients, and existing defense mechanisms, while effective under independent and identically distributed (i.i.d.) settings, often exhibit degraded performance in non-i.i.d. scenarios where client data distributions differ. To overcome this limitation, we propose AMF-CFL, a robust aggregation algorithm specifically designed for federated learning under non-i.i.d. conditions. AMF-CFL reduces the influence of malicious updates through a two-step filtering strategy: it first applies multi-k-means clustering to identify anomalous update patterns, followed by z-score-based statistical analysis to refine the selection of benign updates. Comprehensive evaluations against four untargeted and two targeted attack types demonstrate that AMF-CFL effectively preserves the integrity and robustness of the global model, offering a reliable defense in challenging federated learning environments.

1. Introduction

In recent years, federated learning has gained significant attention as a novel distributed machine learning paradigm [1–5]. Traditional machine learning typically trains models from centralized data. However, in the real-world, data may reside in different institutions like healthcare or financial [2,6,7]. In federated learning, users train local models on private data for a certain number of rounds. The client shares the local model updates with a central server. And the central server aggregates their updates from all clients to update the global model, which is distributed back to the participating clients. This iterative process is repeated multiple times to obtain the final global model [8–10]. The distributed characteristic of federated learning have garnered significant attention, but they also render federated learning highly susceptible to attacks from malicious participants.

In federated learning, the presence of malicious clients poses a significant security concern. Malicious participants exhibiting Byzantine behavior can manipulate the distributed training process and compromise the resulting aggregate model. Even a single Byzantine error can potentially cause a significant threat to the training model [11,12]. Except for the Byzantine model accident, the backdoor attack also poses a threat to the security of the model. Malicious clients can inject a backdoor task without affecting the main task [13–17]. During model aggregation, the backdoor task is injected into the global model. Therefore, the design of robust aggregation algorithms to counter malicious attacks is of paramount importance. Many malicious attacks require injecting malicious factors into malicious updates to amplify the malicious impact, and this is contingent on the number of malicious client segments. To defense these potential attack, existing defenses in federated learning primarily fall into two categories: robust aggregation [18–22], which detects and rejects malicious weights, and authentication defenses [23,24], which involve validation using auxiliary datasets either on the client side or server side. However, authentication defenses may contravene the privacy settings of federated learning and risk the leakage of local client information. In the former category, methods like Krum [19], Trimmed-Mean [20], and Median [20] are well-known for their defense mechanisms relying solely on statistical properties. However, this feature has also led to the development of Byzantine attack algorithms, which are designed for them [25].

Although there are many defense algorithms, the data heterogeneous of federated learning affects the performance of these defense algorithms. Data heterogeneity will cause the update of the model to

* Corresponding author.
E-mail addresses: bowang@dlut.edu.cn (B. Wang), xiaoruidai@dlut.mail.edu.cn (X. Dai), wei.wong@ia.ac.cn (W. Wang), yz2018@mail.dlut.edu.cn (Z. Yang), hailap@mail.dlut.edu.cn (Z. Wang), maozhenzhang@mail.dlut.edu.cn (M. Zhang).

https://doi.org/10.1016/j.jisa.2026.104387

Available online 29 January 2026

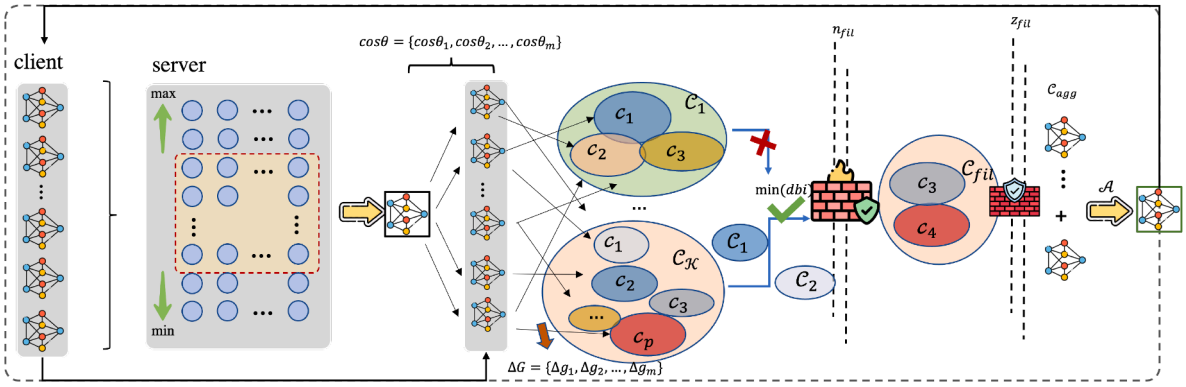


Fig. 1. Framework of AMF-CFL. After local models are uploaded to the server, multiple clustering and two-step filtering are applied on the server side to select benign client updates for aggregation, resulting in a clean global model.

deviate from the optimal update direction or update step. For example, users from different countries may use different languages, so there can be variations in the private data distribution collected by machines in different regions, even for tasks like next-word prediction [9,26] and automatic drive [27]. This is one of our concerns.

To alleviate the performance degradation of federated learning defense algorithm under data heterogeneity. We propose Anomaly Model Filtering based on Clustering in Federated Learning (AMF-CFL). A defense algorithm based on abnormal client filtering for client model parameter clustering. AMF-CFL uses the trimmed global model as a correct or incorrect update direction. This allows us to calculate the cosine similarity to measure angular differences in model updates. This effectively mitigated Byzantine attacks. Subsequently, we employ multiclass classification to filter out potential malicious updates, followed by an investigation of exceptional updates using z-scores. This process yields the final cluster of benign models. We evaluated our approach under non-independent and non-identically distributed data sampling on two datasets, subjecting them to various Byzantine and targeted attacks. The results demonstrate our capability to effectively filter out Byzantine attacks, ensuring the convergence and availability of the global model.

Our summarize the contribution of this paper as follows:

- We conducted research on FL robust aggregation under non-i.i.d data. In the context of non-i.i.d data, our experimental results indicate that it filters malicious clients through clustering and outlier filtering, particularly in the case of attacks with amplification factors.
- We propose to use the potential direction of global model updates and the update gradient of each client as features to distinguish benign clients from malicious clients.
- We validated AMF-CFL using four types of untargeted attacks and two types of targeted attacks. The results demonstrate its effectiveness in filtering out anomalous model updates while ensuring that unfiltered malicious model updates do not have an impact on the attack.

2. Related work

2.1. Federated learning

In federated learning, to learn the globally optimal model w^*, a process of T rounds of iteration is performed. The server S initializes the global model w_0 and sends it to all clients [9]. In the t-th round, S broadcasts the global model w_{t-1} to m clients. Subsequently, the clients locally train their models w_t^i using their private local data D_i, calculate the difference between the local model and the previous global model Δw_t^i, and upload this difference to the server S. In the t-th round, the global model is updated as follows: $w_t = w_{t-1} + \frac{1}{m}\sum_{i=1}^{m} \Delta w_{t-1}^i$.

2.2. Threats from malicious clients

Malicious client attacks are security threats to federated learning. In this paper, we focus on two types of attack. They are untargeted attacks and targeted attacks [28,29].

Untargeted attacks typically assume that malicious clients are ϵ-Byzantine ($0 < \epsilon < 0.5$), meaning that in each round, there are ϵm malicious clients. These malicious clients can manipulate their model updates to achieve their purposes. In Byzantine attacks, they do not need to adhere to the settings of federated learning and have a complete understanding of the system's configuration and learning algorithms. For instance, the TrimmedMean Attack (TMA) [25] bypasses robust aggregation by recording the upper and lower bounds of all model updates. The Inner-product Manipulation Attack attack (IMA) [30] disrupts the defense of robust aggregation by manipulating the inner product of model update gradients. Sign Flipping attacks [23] modify the direction of malicious model updates to disrupt the performance of the global model, and Additive Noise Attacks [23] introduce noisy perturbations to malicious model updates to reduce the availability of the global model.

In targeted poisoning attacks, the objective of malicious participants is to manipulate the global model $\mathcal{M}$ in a way that $\mathcal{M}$ makes specific predictions on certain data, such as predictions on one or two classes. In a Label Flip Attack [23], malicious participants change the training labels from the original labels c_{source} to target labels c_{target}. For any $x_j \in D_j$, where $x_j = (f_j, c_j)$, and $c_j = c_{source}$, it represent the attack scenario as $x_j' = (f_j, c_{target})$. Here, $c_{source} \rightarrow c_{target}$ denotes label flipping attack. We flipped label 0 to label 1 ($0 \rightarrow 1$) on the MNIST dataset and labeled t-shirt to trouser ($t - shirt \rightarrow trouser$) on the Fashion-MNIST dataset. The other is a Model Replacement Attack (MRA) [15]. Malicious clients modify the data of the original training images by adding a black square of size 5×5 and changing the original label of the data to the target label.

2.3. Byzantine-robust FL methods

Currently, there are defense methods based on statistical characteristics, such as Krum [19], which calculates the gradient differences between each client's model update and those of the other $N - f - 2$ clients and selects the smallest difference as the global model update to be distributed to clients. The TrimmedMean method [25], on the other hand, sorts the model update gradients, filters out the β largest and smallest update values, and then aggregates the remaining updates. In addition, Median [25] is an aggregation algorithm based on the median of coordinates, which uses the median of the model update parameters as the global model update. These robust aggregation algorithms select the smallest update based on statistical model differences as the global model update direction. However, the performance of these robust aggregation defenses can degrade in the presence of non-independent and non-identically distributed data. In addition, some attack methods

Algorithm 1 AMF-CFL.

Input:Model parameters w_0, total training round T, number of client N, z-score threshold z_{fil}, TrimmedMean threshold β, $\mathcal{K}$ is Classified quantity
Output: Global model w^*

for each training round t in $[1, T]$ **do**
  for each client i in N **do**
    $w_t^i \leftarrow$ `Local_Update`(w_{t-1})
  end for
  $\mathbf{W}_t = \{w_t^1, \ldots, w_t^N\}$
  $\cos\Theta = \{\cos\theta_1, \ldots, \cos\theta_m\}$
  $C_{fil} \leftarrow$ `k-means-filter`$(\Delta\mathbf{W}_t, \cos\Theta, \mathcal{K})$
  $C_{agg} \leftarrow$ `z-score`$(\Delta\mathbf{W}_t, c^*)$
  $w_{t+1} = w_{t-1} + \frac{1}{|C_{agg}|}\sum_{i=1}^{C_{agg}} \Delta w_t^i$
end for

targeting Krum and TrimmedMean have also been proposed [25]. These attacks make the model under Krum and TrimmedMean defense unable to converge to the optimal point.

3. The proposed method

3.1. Framework

Our proposed AMF-CFL aims to use the TrimmedMean method to determine the update direction and assist in distinguishing between client model updates. Whether the TrimmedMean direction is correct or not, it can always obtain model updates consistent or inconsistent with the trueTrimmedMean update direction. All clients are divided into smaller clusters by multi-clustering. In this way, the common features between different clients are detected and the abnormal clusters are filtered. Then after merging large clusters, malicious updates are filtered through the detection of outliers. The full algorithm is given in Algorithm 1. Fig. 1 is the framwork of AMF-CFL.

3.2. Details of the algorithm

Feature Space and Multi-k-means. We perform aggregation using TrimedMean on gradient updates to obtain the model $\hat{w}$. However, our trimming is stricter, filtering out $\beta = 0.3$, selecting only the middle 40% of model parameters for aggregation to obtain $\hat{w}$. And calculate the cosine similarity $\cos\theta_i$ between $\hat{w}$ and the updates w_i from each client. In the TrimmedMean attack algorithm, whether the attack succeeds or not, it uses the cosine similarity between $\hat{w}$ and w_i for classification to distinguish updates with different directions.

We record the gradient Δw_i for each model update and use $\cos\theta_i$ and Δw_i as the feature space for model updates to perform multi-k-means clustering, where $k \in \mathcal{K}$. We use p to represent the number of clusters clustered by the client. In each clustering $C_{\mathcal{K},\mathcal{K}\subset\mathbb{R}}$, a different number of clusters $c_{p,p\in\mathcal{K}}$ will be obtained. This clustering process results in multiple multiclass sets C_p, where each multiclass set $C_p = \{c_1, \ldots, c_p\}$.

First Filtering and Merging Clusters. Then, we utilize the Davies-Bouldin index (DBI) [31] to evaluate the quality of the multiclass $C_{p,p\in\mathcal{K}}$ and select the optimal classification C^*. DBI is an index to evaluate the clustering effect:

$$\overline{S} = \frac{2}{|c|\cdot(|c|-1)} \cdot \sum_{1\le i\le j\le|c|} dist(x_i, x_j) \tag{1}$$

$$d_{cen}(c_p, c_q) = dist(\mu_p, \mu_q) \tag{2}$$

$$DBI = \frac{1}{\mathcal{K}} \sum_{i=1, p\neq q}^{\mathcal{K}} \max\left(\frac{\overline{S}_p + \overline{S}_q}{d_{cen}(c_p, c_q)}\right) \tag{3}$$

Algorithm 2 Function k-`means filter`.

Input:Model updates gradient $\Delta\mathbf{W_t}$, cosine similarity $\cos\Theta$, multi-k-means $\mathcal{K}$
Output: Clusters after the first filter C_{fil}

$w_t^1 < w_t^2 < \ldots, w_t^N \leftarrow$sorted$(\mathbf{W}_t)$
$w_t^{N-\beta N}, \ldots, w_t^{\beta N} \leftarrow$`prune`$(\beta, [w_t^1, \ldots, w_t^N])$
$\hat{w}_t = \frac{1}{(1-2\beta)N}\sum_{c=1}^{(1-2\beta)N} w_t^c$
$\Delta w_i^t = w_t^i - w_{t-1}^i$, $\Delta\hat{w}_i^t = \hat{w}_t^i - \hat{w}_{t-1}^i$
$\cos\theta_i = \frac{\Delta\hat{w}_t \cdot \Delta w_t^i}{\|\Delta\hat{w}_t\| \cdot \|\Delta w_t^i\|}$
for p in $\mathcal{K}$ **do**
  $C_p \leftarrow$ k-**means**$(p, \Delta\mathbf{W}, \cos\Theta)$
  $C_p = \{c_1, \ldots, c_p\}$
end for
$C = C_1, \ldots, C_p$
$dbi^* \leftarrow \arg\min_{p\in\mathcal{K}}(dbi_p)$
$C^* \leftarrow dbi^*$
if `len`$(c_p) > n_{fil}, c_p \in C^*$ **then**
  $\cos\overline{\theta}_p = \frac{1}{|c_p|}\sum_i^{c_p}\cos\theta_i$
  $\Delta\overline{w}_t^p = \frac{1}{|c_p|}\sum_i^{c_p}\Delta w_t^i$
  d_p is the intra-cluster distance of each cluster
end if
$\Delta\overline{\mathbf{W}}_t, \cos\overline{\Theta}, \mathbf{D}_p$ Represent intra-cluster average gradient, cosine similarity and intra-cluster distance, respectively
$C_1^*, C_2^* \leftarrow$ 2-kmeans$(\Delta\overline{\mathbf{W}}_t, \cos\overline{\Theta}, \mathbf{D}_p)$
$C_{fil} \leftarrow$Select majority cluster C_1^* or C_2^*
return C_{fil}

Algorithm 3 Function `z-score`.

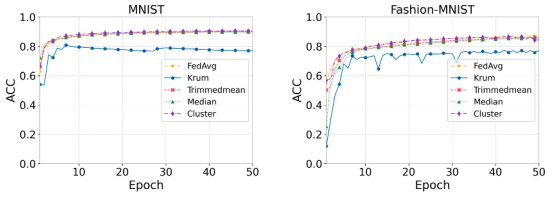
Input:Cluster to be filtered C_{fil}, Model update gradient$\Delta\mathbf{W}$, z-score threshold z_{fil}
Output: Benign client cluster C_{agg}

μ = `mean`(ΔW_t)
σ = `std`(ΔW_t)
for i in C_{fil} **do**
  $z_i = |\frac{\Delta w_t^i - \mu}{\sigma}|$
  if $z_i < z_{fil}$ **then**
    model Δw^i add in C_{agg}
  end if
end for
return C_{agg}

where $dist(\cdot)$ is a function to calculate the distance of two samples, $\overline{S}_p$ is the average distance of the sample in cluster $c_{p,p\in\mathcal{K}}$, $\mu_p = \frac{1}{c_p}\sum_{1<i<|c_p|} x_i$ is the central point in cluster c_p. $d_{cen}(\cdot)$ is a function to calculate the distance between two clusters. We calculate the value of DBI $dbi_{p,p\in\mathcal{K}}$ for different $C_{p,p\in\mathcal{K}}$ by Eqs 1–3. Then the number of clustering times with the lowest DBI value is selected as the target clustering number p^* to obtain C^*

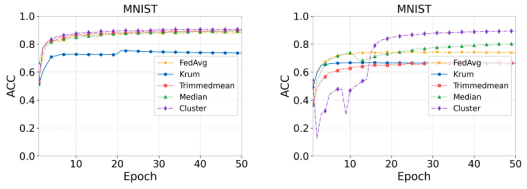
As shown in Algorithm 2, we use the optimal clustering C^* to obtain benign cluster C_{fil}. Then, for each cluster $c_p \in C^*$, filters out the abnormal clusters according to the number of clusters n_{fil}. Calculate the $\cos\overline{\theta}_{i,i\in c_p}, \Delta\overline{w}_{i,i\in c_p}, d_p$ in each cluster as classification feature. The remaining $C^* = \{c_1, \ldots, c_p\}$ are divided into two cluster C_1^* and C_1^* by 2-means. Merge clusters in the same category C_1^* or C_2^*. Select a large number of clusters as benign clusters to be aggregated C_{fil}.

z-score Filtering and Aggregation. The selected cluster C_{fil} obtained in the previous step is considered the benign cluster. We apply z-score filtering to the gradients within this cluster. This filtering can effectively

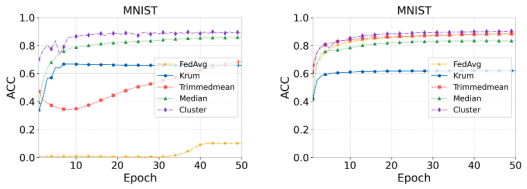


(a) MNIST w/o attack (b) Fashion-MNIST w/o attack

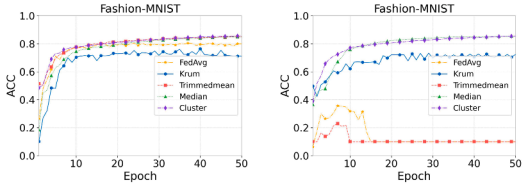
Fig. 2. Robust aggregation without attacks. In each communication round of federated learning, updates from all clients are aggregated.



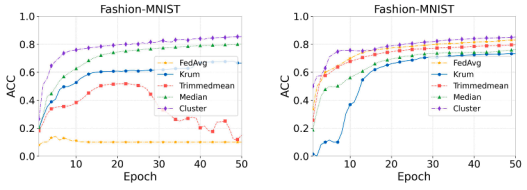
(a) Additive Noise Attack (b) TrimmedMean Attack



(c) Sign Flip Attack (d) Inner Product Manipulation Attack

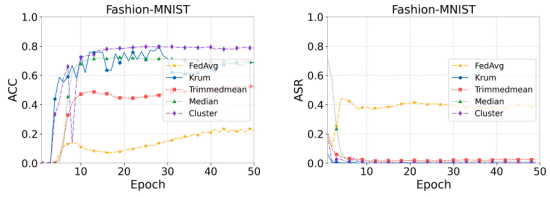


(e) Additive Noise Attack (f) TrimmedMean Attack

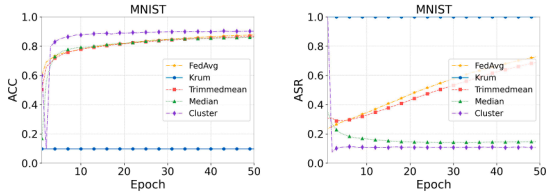


(g) Sign Flip Attack (h) Inner Product Manipulation Attack

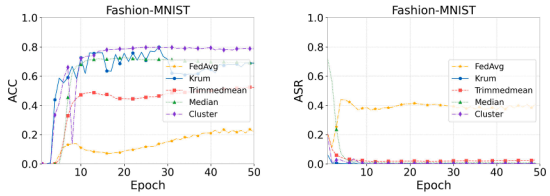
Fig. 3. Robust aggregation under four kinds of non-targeted attacks.



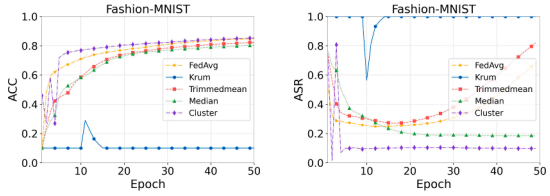
(a) ACC of Label Flip Attack (b) ASR of Label Flip Attack



(c) ACC of Model Replacement Attack (d) ASR of Model Replacement Attack

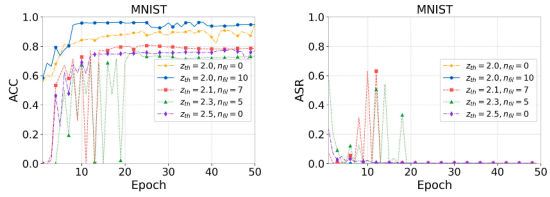


(e) ACC of Label Flip Attack (f) ASR of Label Flip Attack



(g) ACC of Model Replacement Attack (h) ASR of Model Replacement Attack

Fig. 4. Robust aggregation under two kinds of targeted attacks.



(a) ACC under different filtering thresholds (b) ASR under different filtering thresholds

Fig. 5. Effect of gradual relaxation of filtering threshold z_{fil} and n_{fil} on defense effect.

defend against malicious attacks with magnification factors. Malicious attacks that are not filtered have minimal impact on the model due to the constraints on update gradients and their quantity. Finally, we use the aggregation rule $\mathcal{A}$ to aggregate the remaining model updates, resulting in the global optimal model w^*. The pseudo-code of z-score is shown in Algorithm 3.

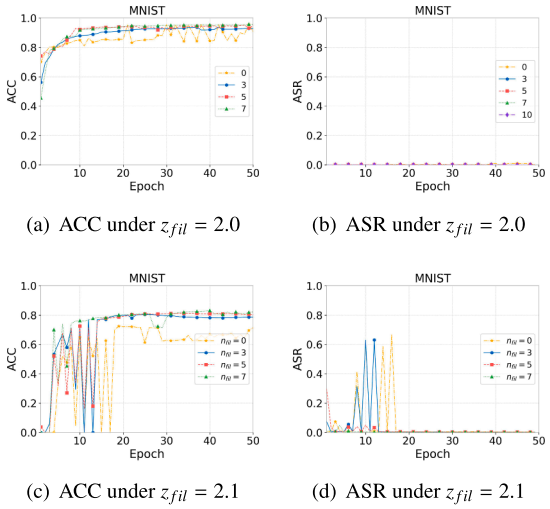
4. Performance evaluation

4.1. Experiment setup

FL setting. We conducted tests on federated learning using two datasets: MNIST and Fashion-MNIST. MNIST consists of 10 different digit labels, ranging from 0 to 9, with a total of 60,000 training images and 10,000 testing images. Fashion-MNIST consists of 10 different fashion categories, such as dresses, shirts, and bags, with a total of 70,000 grayscale images of size 28×28. Out of these, 60,000 images are used for training, and 10,000 are used for testing. In each iteration, we involved 100 clients in the training process, with a malicious client fraction of $\epsilon = 0.3$. This means that in each iteration, there are 30 malicious clients and 70 benign clients. Our model training is performed locally for 5 epochs, and the communication between clients and the parameter server occurred for $R = 50$ rounds. By default, we used the Dirichlet distribution to sample data to create non-i.i.d datasets for training. The parameter α of the

Table 1
Performance comparison of AMF-CFL and baseline robust aggregation methods on the CIFAR-10 dataset under various attack scenarios. The table reports the global model accuracy (%) for each defense method under no attack and six different attacks, including Krum, Trimmed mean, Inner product, Additive noise, Label flip, and Gradient rise. Results demonstrate that AMF-CFL consistently maintains high accuracy across all attack types, highlighting its robustness in federated learning with heterogeneous data.

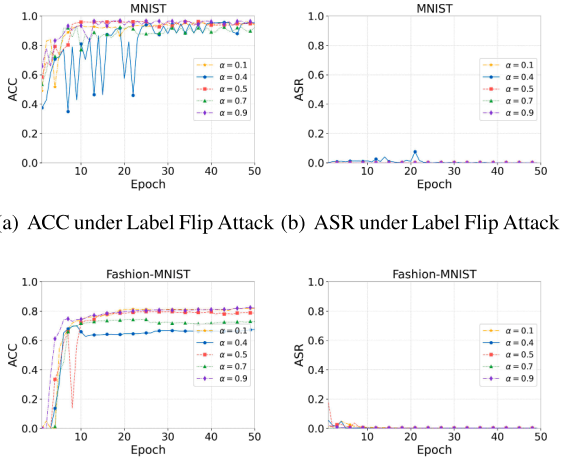
Defense	No Attack	Attack Krum	Attack Trimmedmean	Inner Product	Addition Noise	Label FLIP	Gradient Rise
No Defense	51.65	20.56s	10.00	45.93	24.48	48.89	10.00
Krum	24.47	23.40	25.31	14.45	23.32	24.24	18.44
MKrum	49.28	10.00	48.72	10.00	48.48	46.94	47.51
Trimmedmean	51.30	18.31	10.00	37.03	40.05	51.36	13.93
Median	51.30	23.71	14.59	23.74	49.37	47.65	40.96
Bulyan	51.15	20.09	10.00	41.4	29.55	50.04	10.00
Flare	51.54	9.35	10.20	19.12	34.06	51.12	23.35
AMF-CFL	51.63	50.93	51.68	51.61	51.86	51.14	48.71



(a) ACC under $z_{fil} = 2.0$ (b) ASR under $z_{fil} = 2.0$

(c) ACC under $z_{fil} = 2.1$ (d) ASR under $z_{fil} = 2.1$

Fig. 6. Evaluation of the impact of clustering threshold n_{fil} on robust aggregation across different datasets.



(a) ACC under Label Flip Attack (b) ASR under Label Flip Attack

(c) ACC under Label Flip Attack (d) ASR under Label Flip Attack

Fig. 7. The impact of non-i.i.d. parameters on federated learning performance across communication rounds.

Dirichlet distribution is set to 0.5, where smaller values indicate a more severe non-IID setting. In our filtering setup, the number of filtered updates in the first round is set to $n_{fil} = 10$, and the filtering threshold in the second round is set to $z_{fil} = 2$.

Adversary Setting. In the experiments, we consider six different attack scenarios. Four of them are non-targeted attacks, including the Inner Product Manipulation Attack (IMA), TrimmedMean Attack (TMA), Sign Flip Attack, and Additive Noise Attack. The other are two target attacks, consisting of the Label Flip Attack and Model Replacement Attack (MRA). In the Label Flip Attack, we selected two sets of labels to flip and introduced amplification factors to enhance the attack's effectiveness. In the Model Replacement Attack (MRA), malicious clients added a trigger to the data they possessed. This trigger is a 5×5 pixel black square added to the training data, and the labels are modified to 0. In the Model Replacement Attack (MRA), malicious clients added a trigger to the data they possessed. This trigger is a 5×5 pixel black square added to the training data, and the labels are modified to 0.

Metrics. We use classification accuracy and attack accuracy to measure defense effectiveness. In non-target attacks, only ACC is used to evaluate model availability and quantify the impact of malicious model updates on the global model. For target attacks, we use both ACC and ASR to evaluate the availability of the model. Target attacks are designed for specific backdoor tasks, which means that their goal is to increase ASR as much as possible while maintaining ACC.

4.2. Performance analysis

Defense evaluation without attack. We conduct extensive evaluations on MNIST, Fashion-MNIST, and CIFAR-10. As illustrated in Fig. 2, we compare the performance of AMF-CFL with baseline defenses under various attack settings. AMF-CFL achieves performance on par with FedAvg in the absence of attacks, indicating that the proposed method introduces negligible accuracy degradation in benign scenarios.

Defense evaluation with non-targete attacks We evaluate four representative Byzantine attacks across five robust aggregation rules $\mathcal{A}$ on MNIST and Fashion-MNIST. The results are presented in Fig. 3. Among all methods, Krum exhibits the lowest accuracy, which can likely be attributed to its strategy of selecting only a single client update at each iteration. Without randomly sampling a sufficiently diverse subset of clients, Krum tends to repeatedly select the same clients across iterations, thereby limiting its robustness–especially under non-i.i.d. data distributions. Overall, existing robust aggregation rules are significantly affected by all four Byzantine attacks. In contrast, AMF-CFL effectively mitigates the impact of malicious updates on the global model and consistently achieves the highest accuracy across the evaluated settings. Furthermore, we extend our analysis to CIFAR-10 and include a broader set of robust aggregation methods, as summarized in Table 1. The results show that in the absence of attacks, AMF-CFL attains performance comparable to FedAvg, indicating minimal utility degradation under benign conditions. Under various attack scenarios, AMF-CFL maintains consistently strong performance. One limitation is a slight drop in accuracy under the gradient ascent attack; however, AMF-CFL still outperforms all baseline defense mechanisms by a clear margin, demonstrating its superior robustness.

Defense evaluation with targeted attacks. Fig. 4 presents the performance of the five aggregation rules under two targeted attack settings.

Table 2
Impact of the cluster filtering threshold n_{fil} on model accuracy under gradient rise and additive noise attacks. A moderate value (e.g., $n_{\text{fil}} = 10$) provides the best robustness-accuracy balance.

n_{fil}	3	5	10	20	30
Gradient Rise	48.67	49.91	51.39	51.38	49.20
Additive Noise	51.43	51.00	51.42	48.74	46.15

Table 3
Effect of the z-score threshold z_{fil} on global model accuracy. Stricter thresholds more effectively filter poisoned updates, preventing performance degradation.

z_{fil}	0.5	1.0	1.5	2.1	2.5
Gradient Rise	47.25	51.7	50.41	51.20	49.14
Additive Noise	45.88	50.25	51.67	51.79	51.41

Table 4
Influence of the aggregation balance factor β on the defense performance. Results indicate that the method remains stable across different β settings.

β	0.1	0.2	0.3	0.4	0.5
Gradient Rise	49.91	50.59	51.59	49.99	50.21
Additive Noise	50.41	51.29	51.44	50.33	49.51

The key difference between the two attacks lies in the evaluation metrics: under the label-flipping attack, ACC denotes the accuracy with respect to the original labels, whereas ASR measures the proportion of samples misclassified into the attacker-specified target label. Under the label-flipping attack, the accuracies of FedAvg, Krum, and TrimmedMean with respect to the original labels are substantially degraded, while the target-label accuracies of FedAvg and TrimmedMean even increase due to the injected bias. For the MRA algorithm, Krum fails to converge on the main task–consistent with the earlier observation regarding its repeated client selection–but still yields a high accuracy on the adversarial (backdoor) task. In contrast, AMF-CFL exhibits markedly superior defensive performance against both types of targeted attacks, effectively suppressing unintended target-label predictions while maintaining high accuracy on the original labels.

Impact of Hyperparameters in Cluster-Based Filtering. To investigate the sensitivity of the defense to its hyperparameters, we evaluate the z-score threshold z_{fil}, the cluster filtering threshold n_{fil}, and the aggregation balance factor β. Table 2, Table 3, and Table 4 jointly demonstrate that relaxing either the z-score bound or the cluster-filter count consistently degrades the global model accuracy. This trend aligns with the observations in Fig. 5: looser constraints allow a larger fraction of poisoned updates to bypass the filter, confirming the necessity of stricter filtering. Specifically, setting $z_{\text{fil}} = 2.1$ and $n_{\text{fil}} = 10$ achieves a strong balance between robustness and clean accuracy. Furthermore, when fixing the z-score threshold and varying the cluster filtering depth, the results in Fig. 6 and Table 2 indicate that increasing n_{fil} consistently stabilizes the global model in large-scale FL (100 clients), highlighting the benefit of applying multiple layers of cluster-level filtering. Finally, Table 4 shows that performance remains relatively stable across a wide range of β, demonstrating that the defense is not overly sensitive to the aggregation balance factor.

The influence of non-independent and identical distribution. Fig. 5 illustrates the impact of progressively relaxing the filtering constraints. As the constraints become looser, the performance of the global model gradually degrades, indicating that more malicious updates pass through the filter. This confirms the effectiveness of the filtering mechanism: most malicious updates are successfully removed when the parameters are set to $z_{\text{fil}} = 2.1$ and $n_{\text{fil}} = 10$. The parameter z_{fil} controls the sensitivity of the z-score-based statistical filtering; lower values make the filter stricter, potentially removing some benign updates, while higher values may allow more malicious updates to pass. Therefore, z_{fil} should be chosen to balance robustness and model accuracy.

Fig. 6 further examines the influence of varying n_{fil}, the number of cluster-based filtering layers, while fixing z_{fil}. In a setting with 100 participating clients, the results show that performance consistently improves as n_{fil} increases, highlighting the advantage of multiple filtering layers in large-scale federated learning. Increasing n_{fil} enhances the detection of anomalous updates by capturing diverse abnormal patterns across clusters; however, excessively large values may introduce computational overhead without substantial performance gains. Based on our experiments, a range of $n_{\text{fil}} = 5$-15 is generally effective, depending on the number of clients and the heterogeneity of their data distributions (Fig. 7).

5. Conclusion

In this paper, we conduct a study on several aggregation algorithms $\mathcal{A}$ in Federated Learning (FL). We test and evaluate these algorithms against model replacement attacks, label flip attacks, and four Byzantine attacks. We design a Federated Learning filtering algorithm based on multi-class classification. This algorithm aims to filter potential malicious attacks in FL with non-identically distributed data, resulting in benign update clusters. Our algorithm can also serve as a pre-processing module for robust aggregation, eliminating data distribution anomalies or malicious clients. Subsequently, the benign clusters obtained can be robustly aggregated using aggregation algorithms. Through extensive experiments on two datasets, we demonstrated the effectiveness of our filtering algorithm in preserving model availability.

CRediT authorship contribution statement

Bo Wang: Project administration; **Xiaorui Dai:** Methodology; **Wei Wang:** Investigation; **Zi Yang:** Data curation; **Zhaoning Wang:** Formal analysis; **Maozhen Zhang:** Methodology.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ding J, Tramel E, Sahu AK, Wu S, Avestimehr S, Zhang T. Federated learning challenges and opportunities: an outlook. In: ICASSP 2022-2022 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2022, p. 8752–6.
[2] Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol 2019;10(2):1–19.
[3] Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Found Trends Mach Learn 2021;14(1–2):1–210.
[4] Kim J, Kim G, Han B. Multi-level branched regularization for federated learning. In: International conference on machine learning. PMLR; 2022, p. 11058–73.
[5] Sattler F, Müller K-R, Samek W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Trans Neural Netw Learn Syst 2020;32(8):3710–22.
[6] Cheng K, Fan T, Jin Y, Liu Y, Chen T, Papadopoulos D, et al. Secureboost: a lossless federated learning framework. IEEE Intell Syst 2021;36(6):87–98.
[7] Nagalapatti L, Mittal RS, Narayanam R. Is your data relevant?: dynamic selection of relevant data for federated learning. In: Proceedings of the AAAI conference on artificial intelligence; vol. 36. 2022, p. 7859–67.
[8] Kulkarni V, Kulkarni M, Pant A. Survey of personalization techniques for federated learning. In: 2020 fourth world conference on smart trends in systems, security and sustainability (worlds4). IEEE; 2020, p. 794–7.
[9] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. PMLR; 2017, p. 1273–82.

[10] Gao D, Yao X, Yang Q. A survey on heterogeneous federated learning. arXiv Preprint arXiv:221004505 2022.
[11] Rodríguez-Barroso N, Jiménez-López D, Luzón MV, Herrera F, Martínez-Cámara E. Survey on federated learning threats: concepts, taxonomy on attacks and defences, experimental study and challenges. Infus 2023;90:148–73.
[12] Nair AK, Raj ED, Sahoo J. A robust analysis of adversarial attacks on federated learning environments. Comput Stand Interfaces 2023;:103723.
[13] Gong X, Chen Y, Wang Q, Kong W. Backdoor attacks and defenses in federated learning: state-of-the-art, taxonomy, and future directions. IEEE Wirel Commun 2022.
[14] Cheng S, Liu Y, Ma S, Zhang X. Deep feature space trojan attack of neural networks by controlled detoxification. In: Proceedings of the AAAI conference on artificial intelligence; vol. 35. 2021, p. 1148–56.
[15] Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: International conference on artificial intelligence and statistics. PMLR; 2020, p. 2938–48.
[16] Kaviani S, Shamshiri S, Sohn I. A defense method against backdoor attacks on neural networks. Expert Syst Appl 2023;213:118990.
[17] Shejwalkar V, Houmansadr A, Kairouz P, Ramage D. Back to the drawing board: a critical evaluation of poisoning attacks on production federated learning. In: 2022 IEEE symposium on security and privacy (SP). IEEE; 2022, p. 1354–71.
[18] Chen Y, Su L, Xu J. Distributed statistical machine learning in adversarial settings: byzantine gradient descent. Proc ACM Meas Anal Comput Syst 2017;1(2):1–25.
[19] Blanchard P, El Mhamdi EM, Guerraoui R, Stainer J. Machine learning with adversaries: byzantine tolerant gradient descent. Adv Neural Inf Process Syst 2017;30.
[20] Yin D, Chen Y, Kannan R, Bartlett P. Byzantine-robust distributed learning: towards optimal statistical rates. In: International conference on machine learning. PMLR; 2018, p. 5650–9.
[21] Sattler F, Müller K-R, Wiegand T, Samek W. On the byzantine robustness of clustered federated learning. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2020, p. 8861–5.
[22] Kasyap H, Tripathy S. Beyond data poisoning in federated learning. Expert Syst Appl 2024;235:121192.
[23] Li S, Cheng Y, Wang W, Liu Y, Chen T. Learning to detect malicious clients for robust federated learning. arXiv Preprint arXiv:200200211 2020.
[24] Cao X, Fang M, Liu J, Gong NZ. Fltrust: byzantine-robust federated learning via trust bootstrapping. arXiv Preprint arXiv:201213995 2020.
[25] Fang M, Cao X, Jia J, Gong N. Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX security symposium (USENIX security 20). 2020, p. 1605–22.
[26] Shakhovska K, Dumyn I, Kryvinska N, Kagita MK. An approach for a next-word prediction for ukrainian language. Wire Commun Mobile Comput 2021;2021:1–9.
[27] Milani S, Khayyam H, Marzbani H, Melek W, Azad NL, Jazar RN. Smart autodriver algorithm for real-time autonomous vehicle trajectory control. IEEE Trans Intell Transp Syst 2020;23(3):1984–95.
[28] Fang M, Nabavirazavi S, Liu Z, Sun W, Iyengar SS, Yang H. Do we really need to design new Byzantine-robust aggregation rules? arXiv Preprint arXiv:250117381 2025.
[29] Wang N, Xiao Y, Chen Y, Hu Y, Lou W, Hou YT. Flare: defending federated learning against model poisoning attacks via latent space representations. In: Proceedings of the 2022 ACM on Asia conference on computer and communications security. 2022, p. 946–58.
[30] Xie C, Koyejo O, Gupta I. Fall of empires: breaking byzantine-tolerant sgd by inner product manipulation. In: Uncertainty in artificial intelligence. PMLR; 2020, p. 261–70.
[31] Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Patt Anal Mach Intell 1979;(2):224–7.