Expert Systems With Applications 300 (2026) 130415

Contents lists available at ScienceDirect

Expert Systems With Applications

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

# FLAB: Exploring anomaly bias in backdoor attacks

Hua Wang [a,*], Shaoxiong Wang [a], Lianhua Wang [b], Rui Wang [c]

[a] School of Computer Science, Qufu Normal University, Rizhao, 276500, China
[b] Experimental Teaching and Equipment Management Center, Qufu Normal University, Rizhao, 276826, China
[c] EEMCS, Delft University of Technology, Delft, 2628XE, Netherlands

ARTICLE INFO

*Keywords:*
Deep learning
Federated learning
Byzantine-robust
Backdoor attack
Security

ABSTRACT

Federated learning (FL) allows multiple parties to collaboratively train machine learning models by uploading model updates instead of raw data, thereby protecting data privacy and reducing communication overhead. However, the open nature of public networks makes them vulnerable to attacks. By injecting poisoned samples with backdoor triggers during training and uploading malicious updates, an attacker can manipulate the global model to produce any specified target label. Existing defenses against backdoor attacks have limitations, such as high attack success rates or the need to know or restrict the number of compromised clients controlled by the attacker. To address these shortcomings, we propose FLAB, a novel defense to filter out malicious updates. Specifically, we introduce the concept of anomaly bias to characterize each model update and propose a detection mechanism to quantify their anomalous degrees. By clustering anomaly biases and iteratively reducing the size of the cluster, the anomaly bias associated with the attacker is identified. Finally, all updates with this bias are considered malicious and removed. We conduct exhaustive evaluations of FLAB. Experimental results demonstrate that, compared to existing defenses, FLAB achieves comparable model accuracy while significantly reducing attack success rates. Furthermore, FLAB maintains robust performance even when the number of compromised clients exceeds 80 %.

## 1. Introduction

Federated learning (FL) (McMahan et al., 2017; Zhang et al., 2021) is a promising distributed machine learning framework, particularly well-suited for scenarios with high data privacy and security requirements, such as the Internet of Things (IoT) (Lv & Song, 2019; Nguyen et al., 2021a), finance (Long et al., 2020), and healthcare (Rieke et al., 2020). For example, Google's Gboard (Hard et al., 2018) keyboard optimizes its predictive typing model through FL without the need to collect users' input data. As more devices participate, Gboard can better predict users' commonly used words and phrases, correct spelling errors, and even adapt to the input habits of different languages and dialects. Typically, FL trains models on users' devices locally. After a period of training, each device generates a model update (such as gradients or trained models) that does not contain the user's raw input data, and then sends it to the high-performance server. Finally, the server aggregates these updates to form a global model, which is then distributed back to each device to replace its local model. This iterative process continues until the global model converges or the preset number of iterations is reached.

However, FL is highly vulnerable. Attackers can easily generate malicious updates to interfere with the convergence and reduce the accuracy of the global model, referred to as untargeted attacks (Baruch et al., 2019; Biggio et al., 2012; Cao & Gong, 2022; Fang et al., 2020; Rubinstein et al., 2009; Shejwalkar & Houmansadr, 2021; Steinhardt et al., 2017). Alternatively, they can control the output of the global model by training local models using data with backdoor triggers, referred to as targeted attacks or backdoor attacks (Bagdasaryan et al., 2020; Baruch et al., 2019; Bhagoji et al., 2019; Chen et al., 2017a; Gu et al., 2017; Nguyen et al., 2020; Shafahi et al., 2018; Shen et al., 2016; Wang et al., 2020; Xie et al., 2019). In backdoor attacks, once backdoors is successfully implanted into the global model by an attacker, it can cause serious consequences for real-world applications. For instance, in autonomous driving systems (Levinson et al., 2011), backdoor attacks may cause vehicles to make dangerous driving decisions under certain circumstances, endangering the safety of passengers and pedestrians (Han et al., 2022).

Existing defenses against backdoor attacks can be mainly divided into two categories: similarity-based (Fung et al., 2020; Muñoz-González et al., 2019; Nguyen et al., 2022; Sattler et al., 2020b; Wan et al., 2023) and statistics-based (Blanchard et al., 2017; Chen et al., 2017b; Chu et al., 2022; Guerraoui et al., 2018; Krauß & Dmitrienko, 2023; Nguyen et al., 2022; Yang et al., 2019; Yin et al., 2018) anomaly detection methods. The former calculates the cosine similarity between model

* Corresponding author.
*E-mail addresses:* wanghua@qfnu.edu.cn (H. Wang), wsx@qfnu.edu.cn (S. Wang), wanglh@qfnu.edu.cn (L. Wang), r.wang-8@tudelft.nl (R. Wang).

https://doi.org/10.1016/j.eswa.2025.130415
Received 14 August 2024; Received in revised form 4 November 2025; Accepted 11 November 2025
Available online 15 November 2025

updates, penalizing or removing those with abnormal similarity. These defenses are particularly effective against malicious updates that are significantly different from benign ones. However, since attackers can exploit their understanding of the global model's behavior in either the parameter or feature space, they are able to align backdoor updates with benign ones, rendering them difficult to distinguish. Moreover, because clients' data are typically non-IID, the cosine similarity among benign updates tends to be low, which further weakens the discriminative power of such detection methods. The latter category attempts to identify malicious updates through statistical methods such as Euclidean distance, mean or median. However, a major limitation of such methods is that they often rely on prior knowledge or explicit assumptions about the number or proportion of attackers when designing algorithms or establishing theoretical upper bounds. In practice, clients in real-world scenarios are dynamic and participate randomly, and the server usually cannot accurately determine the number of malicious participants in advance. Moreover, attackers may deliberately introduce small but consistent updates to evade detection metrics, thereby biasing the estimation of attacker proportions.

Recent studies have proposed more advanced backdoor attacks (Liu et al., 2024; Zhang et al., 2025), which optimize both the generation and embedding of triggers to enhance stealthiness while preserving attack effectiveness. FLAB explicitly accounts for the number of malicious clients and the intensity of attacks during detection, allowing it to remain effective even against such backdoor attacks. On the defense side, several novel countermeasures have also been introduced (Ding et al., 2025; Yuan et al., 2025). Although these methods achieve strong robustness through sophisticated module designs, their core detection mechanisms still rely on clustering or similarity-based measures, which may limit their adaptability in heterogeneous or highly non-IID FL settings.

To this end, we propose FLAB, a novel defense to address these gaps. Our approach differs fundamentally from prior work in its underlying objective. Previous defenses primarily aimed to design more robust aggregation algorithms that mitigate effects of malicious updates on the global model - a relatively mild and generic strategy that can also handle untargeted or noisy updates. However, such generality often comes at the cost of specificity, making these methods less effective against backdoor attacks. In contrast, FLAB analyzes the attack targets of backdoor attackers to enable a target-oriented and fine-grained identification of backdoor updates rather than merely mitigating their effects, thereby enhancing overall defense effectiveness. To achieve this goal, FLAB explicitly models the anomaly bias introduced by backdoor updates by analyzing how these updates distort the model's parameter space. Unlike existing methods, FLAB does not rely on geometric similarity or global clustering assumptions. Instead, it infers common attack targets by leveraging the frequency and magnitude of anomaly bias occurrences in client-side updates-essentially, the labeling behavior within locally trained client batches. This frequency-driven target inference, combined with an iterative strategy that removes candidate biases in ascending order of magnitude, enables FLAB to detect collaborative, low-amplitude backdoor injections without requiring any prior knowledge of the number of attackers.

Specifically, to successfully implant backdoors, an attacker control multiple compromised clients to carry out backdoor attacks, and all of them have a common attack target. Given this observation, we introduce an anomaly bias for each update, where the degree and index represent the proportion and value of the most frequent label in the training batch, respectively. Then, certain anomaly biases that may be related to the attacker are identified based on their anomalous degrees. Finally, by counting the occurrences of each type of anomaly bias across all model updates, FLAB selects the index of the anomaly bias with the most occurrences as the attacker's target label and removes all updates associated with it. This process aims to determine which anomaly bias is most likely related to the attacker by considering frequency of occurrence. Furthermore, if this maximum value is not unique, candidate anomaly biases will be removed one by one according to the order of their degrees from small to large until a unique maximum value is found. While we couldn't be certain that all removed updates are malicious, if they are, this backdoor attack carried out by compromised clients would effectively leave backdoors in the global model.

Our contributions are as follows:

- We explore the essence of backdoor attacks and introduce the concept of anomaly bias to characterize model updates. The anomaly bias for each update reflects the proportion of its corresponding label in the training dataset, enabling defenses to effectively identify backdoor model updates. Additionally, we design a detection mechanism to quantify these biases.
- Considering both the attacker's strength and the attack target, we propose a novel defense aimed at reducing the success rates of backdoor attacks, which analyzes all anomaly biases to determine which one is most likely associated with the attacker and removes all model updates with it.
- We conduct a detailed experimental evaluation of FLAB. Experimental results show that, compared to other defenses, FLAB can effectively defend against backdoor attacks without any auxiliary knowledge, even when the number of compromised clients reaches a staggering 80 % or more.

**Organization.** The rest of the paper is organized as follows: Section 2 reviews the background and related works. Section 3 discusses the threat model. Section 4 introduces the principles of FLAB. Section 6 presents and discusses the experimental results. Finally, we conclude the paper in Section 8.

## 2. Background and related work

### 2.1. Federated learning

FL is deployed in a distributed network with $N$ clients, which can join or leave at any time. Each client $n \in \{1, \dots, N\}$ locally maintains a dataset $D_n$ containing $|D_n|$ data samples, and each sample consists of feature vector $x \in R^d$ and label $y \in \{1, \dots, M\}$, where $d$ and $M$ represent the dimension of the vector and the total number of sample categories in the dataset, respectively. During train iteration $t \in \{1, \dots, T\}$, the server randomly selects $S (0 < S < N)$ clients from the $N$ clients and sends the global model $G^t$ to them. Each selected client $s \in \{1, \dots, S\}$ then randomly samples a batch from its dataset $D_s$, and inputs the feature matrix $X_s \in R^{d \times b}$ into the model through forward propagation, where $b$ represents the batch size. Next, the update $g_s = \nabla f(G^t, X_s; Y_s)$ is calculated through backpropagation, where $f$ represents the loss function (e.g., Mean-Squared Error loss (Kim et al., 2021) or Cross-Entropy loss (Mao et al., 2023)), and $Y_s \in \{1, \dots, M\}^b$ is the label vector corresponding to the input matrix $X_s$. The purpose of applying a loss function is to minimize the difference between the model's predictions and the actual labels. Finally, the server receives and aggregates model updates from all selected clients to update the global model $G^{t+1} = G^t - \sum_{s=0}^{S} w_s g_s$, where $w_s$ represents the weight assigned to $g_s$ based on the aggregation rules.

### 2.2. Backdoor attacks

Backdoor attacks are a type of malicious attack method against neural network models, causing them to consistently produce the same errors on specific types of samples. Backdoor attacks can be categorized into pixel-pattern backdoors (Gu et al., 2017; Xie et al., 2019) and semantic backdoors (Bagdasaryan et al., 2020). In pixel-pattern backdoor attacks, attackers typically make small modifications to training samples, such as adding pixel patterns (called backdoor triggers) to the edges of images. These poisoned samples are used to locally train the models on compromised clients' devices, along with the target labels assigned by attackers. In contrast, in semantic backdoor attacks, attackers

select samples with the same distinctive features from all compromised clients' training dataset as poisoned samples and flip their labels to the target label for training models. Furthermore, attackers can also adjust the parameters of malicious model updates to maximize backdoor impact (e.g., model replacement (Bagdasaryan et al., 2020)) or to avoid detection (e.g., train-and-scale (Bagdasaryan et al., 2020), constrain-and-scale (Bagdasaryan et al., 2020) or LIE (Baruch et al., 2019)).

### 2.3. Related work

Many researchers have been committed to designing defenses based on statistical methods. For example, authors in Yin et al. (2018) removed the $m$ largest and smallest values of each weight and bias in updates, then aggregated the remaining parameters to update the global model, where $m$ represents the number of compromised clients. However, this method requires knowing exactly how many compromised clients there are. Authors in Blanchard et al. (2017) calculated the sum of Euclidean distances between each update and its nearest $n-m-2$ updates, and used the update with the smallest sum as the global update. Although it does not require knowing the exact number of compromised clients, this method limits their number, i.e., $m < (n-2)/2$. Authors in Guerraoui et al. (2018) combined Multi-Krum (Blanchard et al., 2017) and Trimmed-mean (Yin et al., 2018) to remove malicious updates with anomalous parameters that are still geometrically similar to benign updates. However, it requires $m \leq (n-3)/4$, imposing a stricter limit. Authors in Yin et al. (2018) and Chu et al. (2022) assumed that the median of parameters in each dimension is reliable, and excluded or penalized other parameters. Nevertheless, this type of method can only achieve ideal performance when $m < n/2$. Authors in Sun et al. (2019) found that known backdoor attacks generally lead to larger update norms and demonstrated that norm clipping can effectively reduce the success rate of such attacks. However, compromised clients sharing the same attack target can launch effective attacks without employing the model replacement paradigm, and the norms of their updates do not appear abnormal.

Unlike using statistical methods, some defenses exploit the similarity between model updates. Authors in Cao et al. (2020) calculated the cosine similarity between trusted updates on the server and updates uploaded by clients, and removed updates with lower similarity during the aggregation stage. However, this method requires a trusted dataset on the server. Authors in Wan et al. (2023) and Fung et al. (2020) assumed that there is a higher degree of similarity between malicious updates than benign updates, and removed or weakened updates that are highly similar to each other. While this idea is correct, a single powerful attacker can still effectively implant backdoors (Bagdasaryan et al., 2020). Authors in Sattler et al. (2020b) clustered all updates (Sattler et al., 2020a) and retained only the cluster with the most updates. However, this approach is not suitable for situations where malicious updates are in the majority. Authors in Muñoz-González et al. (2019) first computed the weighted average of all updates and then calculated the cosine similarity between this value and each update. Finally, updates with anomalous similarity were discarded. However, for backdoor updates, the difference in similarity is not sufficient to accurately distinguish them. Authors in Nguyen et al. (2022) used HDBSCAN (Campello et al., 2013) to cluster the similarity matrix of updates and clipped all updates based on the Euclidean distance between the median of updates and the global model. However, this method may also inherit the limitations of the two kinds of defenses mentioned above. Authors in Nguyen et al. (2021b) first used dynamic clustering to identify and remove potentially poisoned model updates that have high attack impact, and then eradicated residual backdoors through clipping, smoothing, and noise addition. However, the scheme is computationally intensive.

In addition, some defenses explore new areas but they still have certain limitations. Authors in Andreina et al. (2021) proposed an anti-backdoor scheme based on participant feedback, which verifies the global model's integrity by monitoring changes in the misclassification rates of specific classes. The approach identifies potential backdoors and determines whether to accept the current model accordingly. However, this approach may slow down the iteration process of FL. Authors in Wu et al. (2020) proposed a distributed pruning strategy that determines the global pruning order and rate by aggregating local pruning lists from all participants, combined with feedback from a verification set. However, each training process requires multiple rounds of communication. Authors in Li et al. (2020) used a variational autoencoder (VAE) (Kingma & Welling, 2013) to detect outliers in all model updates. However, the server needs to set up a dataset for training the encoder and decoder, which requires additional time to initiate FL. Authors in Zhang et al. (2022) predicted the direction of each update, and then removed updates with abnormal behaviors. However, this method requires maintaining an attacker-free environment in the early stages of training.

## 3. Threat model

### 3.1. Attack model

Building on previous works (Bagdasaryan et al., 2020; Nguyen et al., 2022; Wan et al., 2023), we construct a more powerful attacker, denoted as $\mathscr{A}$ from the following three perspectives:

**Goals.** $\mathscr{A}$ aims to improve the accuracy of the global model on samples with backdoor triggers without significantly compromising the accuracy on clean samples. Specifically, $\mathscr{A}$ controls all compromised clients to carry out backdoor attacks using the same target label. Additionally, to avoid being detected by the server as much as possible, $\mathscr{A}$ should balance the differences between malicious updates and benign ones.

**Capabilities.** $\mathscr{A}$ can control any number of benign clients or inject any number of fake clients into FL networks. More importantly, $\mathscr{A}$ is free to choose which clients to compromise, ensuring that in each train iteration, the clients selected by the server include all compromised clients.

**Knowledge.** For compromised clients, including fake clients, $\mathscr{A}$ has full access, including the ability to arbitrarily modify samples on their local devices and settings (e.g., loss function, learning rate and batch size). For benign clients, $\mathscr{A}$ can only obtain model updates they upload to the server, which helps $\mathscr{A}$ to generate covert malicious updates similar to benign ones.

### 3.2. Defense model

Our goal is to design a defense method that enhances the global model's accuracy on clean samples while minimizing backdoor attack success rates. Next, we discuss the architectures, application scenarios, and settings of the defense.

**Application scenarios.** In a general neural network (LeCun et al., 2015), layers close to the output are used to extract high-dimensional features of the sample. In other words, these parameters have a greater impact on the model outputs. Typically, for models applicable to classification tasks, the output layer has neurons corresponding to the number of classifications used to produce the model's predictions. The defense takes advantage of this property by using the parameters of the output layer to identify malicious updates, with good scalability and generality for classification tasks.

**Architectures.** The defense filters malicious model updates by analyzing the distribution of model update parameters, comprising an anomaly bias detection mechanism and a compromised client identification strategy. The former performs anomaly parameter detection to detect potentially malicious model updates, while the latter identifies compromised clients based on their characteristics in backdoor attacks. Finally, the server discards all identified malicious updates during the global model update phase.

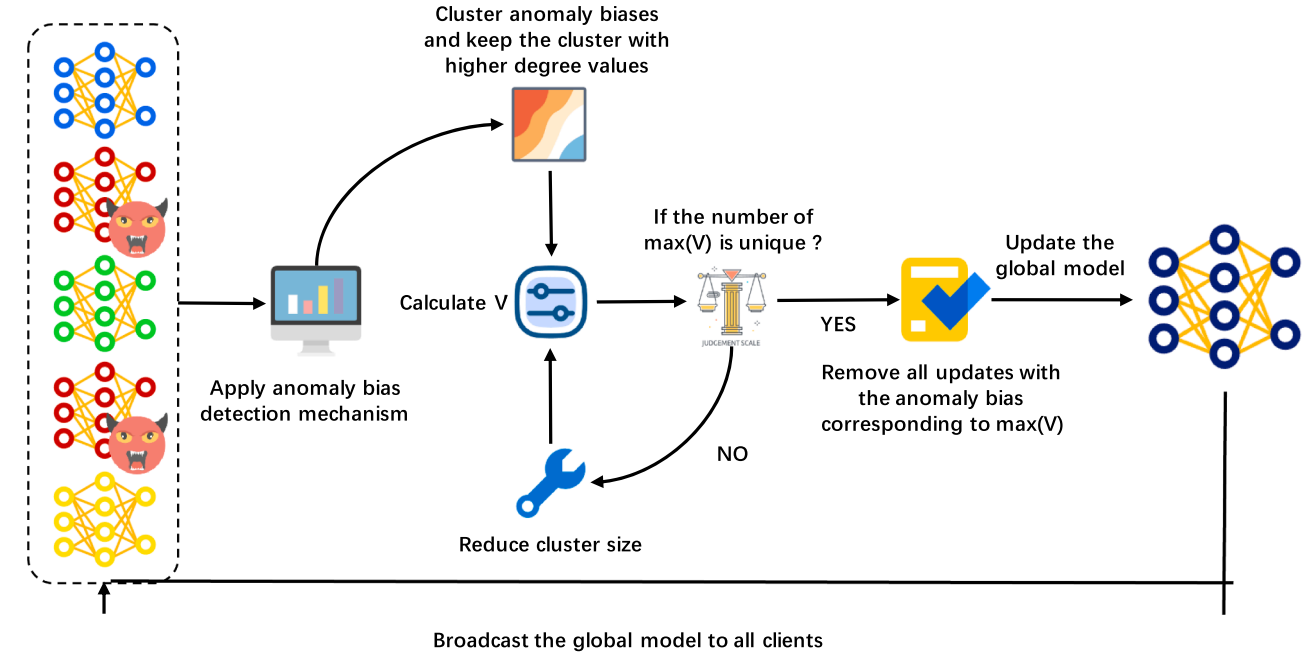



**Fig. 1.** The overview of FLAB.

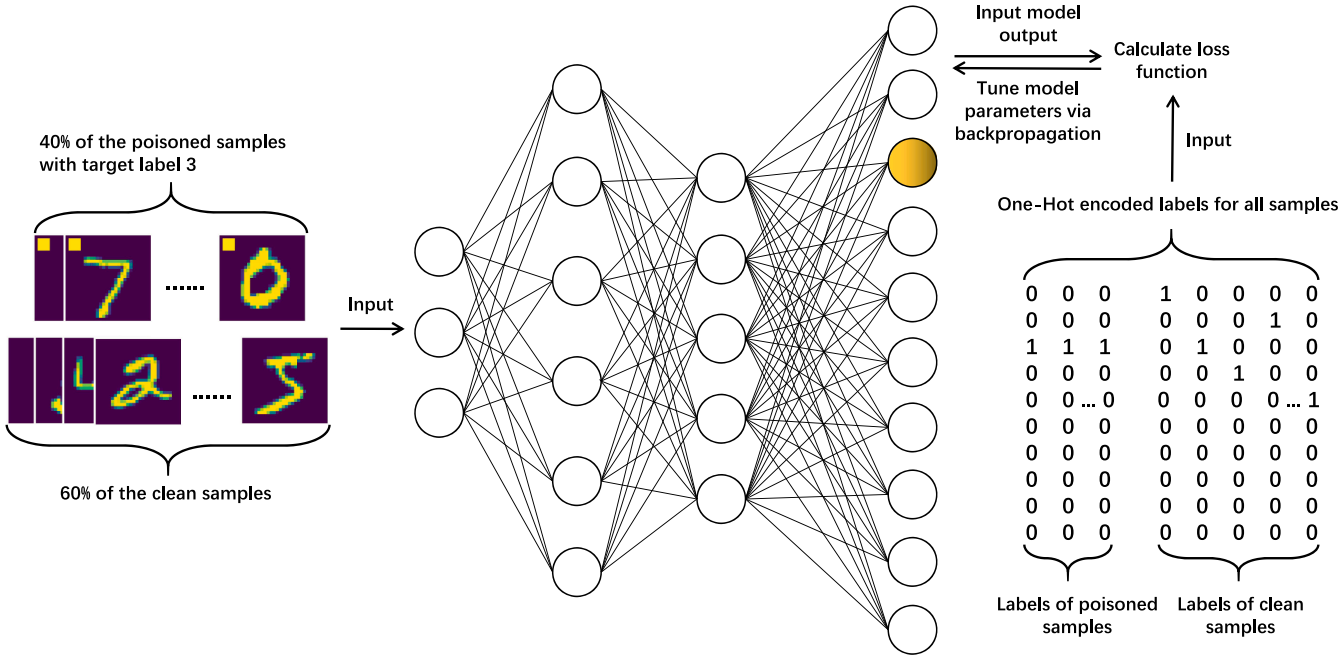



**Fig. 2.** The training process of a compromised client. The training samples contain 40 % of poisoned samples with the same target label assigned by the attacker, causing the model parameters (in the orange neuron) to be adjusted to larger values during training to optimize the prediction effect for this category.

**Settings.** We consider the defense to be performed on the server side. The server has full access to the model updates uploaded by all clients, as well as the global model. In addition, all client-related information is transparent to the server, including local training datasets, the number of compromised clients, and all local training settings, etc.

## 4. Methodology

In this section, we first describe anomaly bias in detail and then propose a detection mechanism based on their anomalous degree. Finally, we eliminate malicious model updates by clustering anomaly biases and applying the majority rule.

### 4.1. Anomaly bias

In neural networks (LeCun et al., 2015), to enable the global model to learn the backdoor task while maintaining accuracy on the main task (Bagdasaryan et al., 2020), the attacker typically embeds triggers into only a portion of the samples, rather than all of them. This approach results in noticeable alterations in the updates they upload.

Specifically, each layer of a neural network is composed of weights $w$ and biases $b$:

$$z^{(l)} = w^{(l)} a^{(l-1)} + b^{(l)}, \tag{1}$$

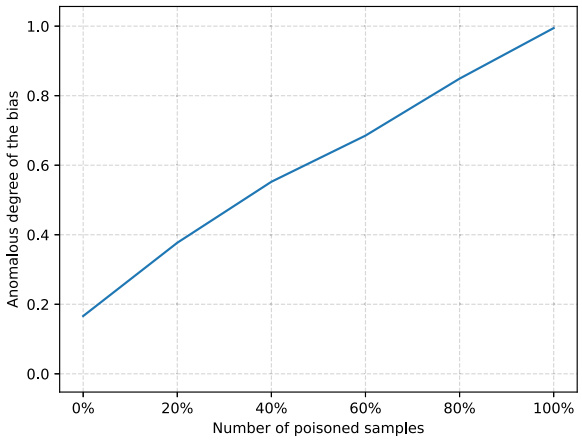



**Fig. 3.** Changes in the degree of an anomaly bias with the proportion of poisoned samples.

where $l \in \{1, \dots, L\}$ and $a$ represent the current layer number and $z^{(l-1)}$ of the applied activation function, respectively. Furthermore, $a^{(L)}$ is the output of the model. For example, in multi-classification tasks, the probability of the model outputting sample category $m \in \{1, \dots, M\}$ is:

$$a_m^{(L)} = Softmax(z_m^{(L)}) = \frac{e^{z_m^{(L)}}}{\sum_{i=1}^{M} e^{z_j^{(L)}}}, \tag{2}$$

where $a_m^{(L)}$ and $z_m^{(L)}$ represent the $mth$ component of $a^{(L)}$ and $z^{(L)}$, respectively.

When there are more training samples with category $m$, the model tends to increase $a_m^{(L)}$ to enhance the prediction probability for this category in order to adapt to this distribution. Furthermore, to indirectly increase $a_m^{(L)}$ via $z_m^{(L)}$, the model adjusts $w^{(L)}$ and $b^{(L)}$ using negative gradients in gradient descent, which leads to an increase in $z_m^{(L)}$. As shown in Fig. 2, during local training on a compromised client, 40 % of the poisoned samples with the same target label make the model tend to increase the parameters of the orange neuron.

Although both $w_m^{(L)}$ and $b_m^{(L)}$ influence the value of $z_m^{(L)}$, $b_m^{(L)}$ directly adds to $z_m^{(L)}$ as a bias term, making its impact more straightforward. Consequently, $b_m^{(L)}$ tends to more directly reflect the magnitude of $z_m^{(L)}$. On the other hand, when calculating gradients, $w_m^{(L)}$ is multiplied by $z_m^{(L-1)}$, which introduces an additional scaling factor. This causes the gradient of $w_m^{(L)}$ to typically be smaller than that of $b_m^{(L)}$, and the update amplitude for $w_m^{(L)}$ is reduced, which is not conducive to calculating the anomalous degree. In summary, we only use $b^{(L)}$ with fewer parameters to complete the subsequent algorithm, and in $b^{(L)}$, we refer to the anomalous component caused by the label distribution as the anomaly bias.

### 4.2. Anomaly bias detection mechanism

The anomalous degree of the update uploaded by client $s$ to the server during training iteration $t$ is quantified as follows:

$$d_s^t = |min(b_s^{(L),t}) - median(b_s^{(L),t} \backslash min(b_s^{(L),t}))|. \tag{3}$$

The degree of anomaly bias increases with the proportion of poisoned samples in the batch, as shown in Fig. 3. Since the server can calculate $d^t$ for each client, it is not a conclusive measure for determining whether updates are malicious. In fact, Eq. (3) provides an intuition: if client $s$ is compromised and calculates the model update $g_s^t$ using samples with backdoor triggers during train iteration $t$, then $d_s^t$ reflects the proportion of poisoned samples, and the index of $min(b_s^{(L),t})$ in $b_s^{(L),t}$ indicates the attacker's target label.

To distinguish malicious updates, Section 4.3 is necessary.

### 4.3. Identifying malicious updates

Since we make no assumptions about the attacker's target label, the number of compromised clients, or the strength of attacks, all these factors need to be considered to identify compromised clients.

Specifically, $C^t = \{d_s^t | s = 1, \dots, S\}$ is clustered into two clusters using 2-Means clustering, i.e., $C_\alpha^t$ and $C_\beta^t$, where $S$ represents the collection of clients randomly selected by the server to participate in training in iteration $t$. For corresponding clients in the cluster $C_{\gamma \in \{\alpha,\beta\}}^t$ with higher anomalous degrees, if compromised, their model updates will implant a backdoor in the global model. Therefore, these clients are considered potentially malicious. Next, the anomaly biases in cluster $C_\gamma^t$ are counted to form a vector $V^t$, i.e., $V^t = \{v_m | m = 1, \dots, M\}$, where $M$ is the dimension of $b_s^{(L),t}$. Since the target label is unique, if $max(V^t)$ occurs only once, all updates with the corresponding anomaly bias are considered malicious and removed. Otherwise, a cycle is performed. In each cycle, $C_\gamma^t$ is updated as $C_\gamma^t \backslash \min(C_\gamma^t)$ and $V^t$ is recalculated iteratively until $max(V^t)$ becomes unique. An overview of the FLAB framework is provided in Fig. 1.

Note that not all updates with this anomaly bias are necessarily malicious. However, to minimize backdoor attack success rates, FLAB removes all updates exhibiting this bias, including potentially benign ones. These updates are trained on unbalanced datasets, so their removal does not significantly affect the model's accuracy on the majority category, but reduces overfitting.

We summarize the specific process of FLAB in Algorithm 1.

**Algorithm 1:** FLAB

**Input:** initial global model $G^0$; total number of training iterations $T$.
**Output:** trained global model $G^T$.

**for** $t = 1$ *to* $T$ **do**
  Initialize an empty vector $C^t$.
  Send global model $G^t$ to $N$ clients.
  Randomly select $S$ clients from $N$ clients.
  **for** $s \in S$ **do**
    $g_s^t \leftarrow \bigtriangledown f(G^t, X_s; Y_s)$
    $b_s^{(L),t} \leftarrow g_s^t[-M:]$
    $d_s^t \leftarrow |min(b_s^{(L),t}) - median(b_s^{(L),t} \backslash \min(b_s^{(L),t}))|$
    $C^t \leftarrow$ append$(d_s^t)$
  **end**
  $C_\gamma^t \leftarrow 2 - Means(C^t)$ and only keep the cluster with higher anomalous degrees.
  **while** ***True*** **do**
    Initialize a counting vector $V^t$ with all zeros.
    **for** $d_s^t$ *in* $C_\gamma^t$ **do**
      $V^t$[the index of $min(b_s^{(L),t})$ in $b_s^{(L),t}$ associated with $d_s^t$] $+= 1$
    **end**
    **if** $max(V^t)$ *is unique* **then**
      $S \leftarrow S \backslash$ clients whose anomaly biases are related to $max(V^t)$.
      **break**
    **else**
      $C_\gamma^t \leftarrow C_\gamma^t \backslash \min(C_\gamma^t)$
    **end**
  **end**
  $G^{t+1} = G^t - \alpha \frac{1}{|S|} \sum_{s \in S} g_s^t$
**end**

**Table 1**
Proportion of the target label correctly identified, MA, and BA of FLAB for different proportions of poisoned samples when the attacker conducts BadNets and the target label is 3 on MNIST.

| Proportion of poisoned samples (%) | PCR (%) | | MA | | BA | |
|---|---|---|---|---|---|---|
| | w | b | w | b | w | b |
| 1 | 2 | 23 | 97.53 | 97.55 | 0.57 | 0.48 |
| 2 | 0 | 24 | 97.46 | 97.30 | 0.85 | 0.88 |
| 4 | 7 | 22 | 97.23 | 97.16 | 5.63 | 2.85 |
| 6 | 13 | 39 | 97.24 | 97.37 | 26.60 | 6.18 |
| 8 | 19 | 56 | 97.12 | 97.12 | 49.43 | 1.26 |
| 10 | 36 | 66 | 97.34 | 97.15 | 65.65 | 0.65 |
| 20 | 55 | 83 | 97.06 | 97.02 | 68.15 | 0.39 |
| 30 | 81 | 100 | 97.05 | 97.01 | 6.67 | 0.24 |

## 5. Convergence analysis

As discussed in Section 2.1, the FL system aims to train the optimal global model $G^*$ to solve the following optimization problem:

$$G^* = argmin_G F(G) \triangleq \frac{1}{S}\sum_{s=1}^{S} f_s(G^t, X_s; Y_s), \tag{4}$$

where $G$ is the global model, and $f_s$ is the objective function for client $s$ using its local training samples, $X_s$ and $Y_s$. Next, we show that, under certain assumptions, the difference between the global model learned by FLAB under attacks and the optimal global model $G^*$ is bounded.

**Assumption 1.**
The global loss function $F(G)$ is $\mu$-strongly convex and has L-Lipschitz continuous gradients. That is, for any $G_1, G_2$, we have

$$F(G_1) \geq F(G_2) + \nabla F(G_2)^T(G_1 - G_2) + \frac{\mu}{2}\|G_1 - G_2\|^2, \tag{5}$$

$$\|\nabla F(G_1) - \nabla F(G_2)\| \leq L\|G_1 - G_2\|. \tag{6}$$

**Assumption 2.**
The anomaly detection mechanism in FLAB excludes a subset of malicious clients. Let $\tilde{S}_t \subseteq S_t$ denote the set of selected clients after filtering. The aggregated gradient is

$$\hat{g}^t := \frac{1}{|\tilde{S}_t|}\sum_{s\in\tilde{S}_t} g_s^t. \tag{7}$$

We assume there exists a bounded bias term $b_t$ such that

$$\|\mathbb{E}[\hat{g}^t] - \nabla F(G^t)\| \leq b_t, \tag{8}$$

$$\mathbb{E}[\|\hat{g}^t - \mathbb{E}[\hat{g}^t]\|^2] \leq \frac{\sigma^2}{|\tilde{S}_t|}. \tag{9}$$

**Theorem 1.**
*Under Assumptions 1–2, let the global model be updated as*

$$G^{t+1} = G^t - \alpha\hat{g}^t, \tag{10}$$

*with learning rate $\alpha \leq \frac{1}{L}$. Then, the expected distance to the optimal model $G^*$ satisfies*

$$\begin{aligned}&\mathbb{E}[\|G^{t+1} - G^*\|^2]\\ &\leq (1 - 2\alpha\mu + \alpha^2L^2)\mathbb{E}[\|G^t - G^*\|^2] + \alpha^2(\sigma^2 + b_t^2)\\ &\quad +2\alpha b_t\mathbb{E}[\|G^t - G^*\|].\end{aligned} \tag{11}$$

*Moreover, if $b_t \leq b$ for all $t$, then $G^t$ converges linearly to a neighborhood around $G^*$, with error radius determined by $\alpha^2(\sigma^2 + b^2)$.*

**Proof.** See Appendix A.

## 6. Experimental evaluation

### 6.1. Experimental settings

#### 6.1.1. Datasets and distributions

We evaluate FLAB on three popular datasets for image recognition tasks: MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017) and CIFAR10 (Krizhevsky et al., 2009). Additionally, in line with recent studies (Nguyen et al., 2022), we utilize the Dirichlet distribution (Ferguson, 1973) to partition the datasets and create a local dataset for each client. Unless otherwise specified, the distribution parameter $\alpha$ is set to 1.

#### 6.1.2. Learning configurations.

We set up 100 clients. In each train iteration $t$, 20 of them are randomly selected to participate in training. To more effectively evaluate backdoor attacks, all compromised clients are included in every iteration. For instance, if the number of compromised clients $m$ is 18, only two benign clients are involved in each iteration.

#### 6.1.3. Models and parameter settings

We use a simple LeNet (LeCun et al., 1989) network as the global model for training on MNIST and Fashion-MNIST, and ResNet-18 (He et al., 2016) for CIFAR10. Additionally, we conduct a total of 100 iterations of training for both MNIST and Fashion-MNIST, and 1000 iterations for CIFAR10. The batch sizes for training and testing are set to 128 and 1000, respectively.

#### 6.1.4. Evaluated attacks

**HTB (Bagdasaryan et al., 2020).** HTB is a semantic backdoor attack that assigns attacker-selected labels to certain images with unique styles. We employ the model replacement approach when $m = 1$ and naive approach when $m > 1$.

**BadNets (Gu et al., 2017).** BadNets is a pattern backdoor attack that involves modifying the pixels of training and test images in a special way. We add a 2-pixel white block in the upper left corner of a certain proportion of images in each batch.

**DBA (Xie et al., 2019).** DBA is a distributed pattern backdoor attack in which the images of each compromised client contain only a portion of the backdoor trigger. The trigger we set consists of four rectangular white blocks, each 4 pixels long and 2 pixels wide, placed in the upper left corner of the images, with two blocks positioned at the top and two at the bottom. Each compromised client randomly selects one of these blocks to modify in its images.

**LFA (Bhagoji et al., 2019).** LFA is not a backdoor attack; instead, it reduces the accuracy of the global model on the main task by arbitrarily changing the labels of images. We modify a certain proportion of images from all compromised clients to have the same target label.

#### 6.1.5. Evaluated defenses

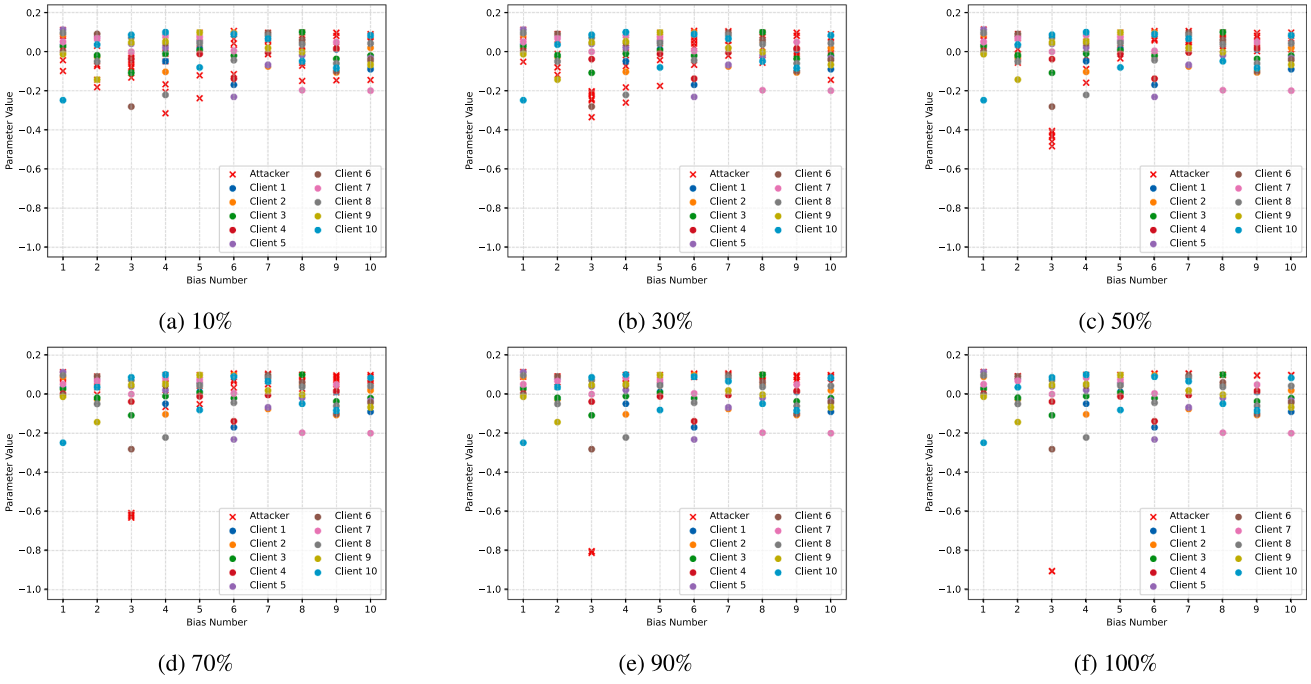
We evaluate several algorithms for comparison with our FLAB, including FLTrust (Cao et al., 2020), FLAME (Nguyen et al., 2022), ARC (Allouah et al., 2024), and MoNNA (Farhadkhani et al., 2023). Additionally, FedAvg (McMahan et al., 2017) is used as the baseline.

#### 6.1.6. Evaluation metrics

**Backdoor Accuracy (BA) .** BA refers to the percentage of samples with backdoor triggers that the global model incorrectly predicts as the target label assigned by the attacker out of the total number of such samples. In each iteration, a higher BA indicates a more effective backdoor attack initiated by the attacker.

**Main Task Accuracy (MA).** MA refers to the percentage of samples that the global model correctly predicts as true labels out of the total number of samples. Generally speaking, backdoor attacks only slightly affect MA because if MA is very low, BA will also be low, which is unacceptable to the attacker.

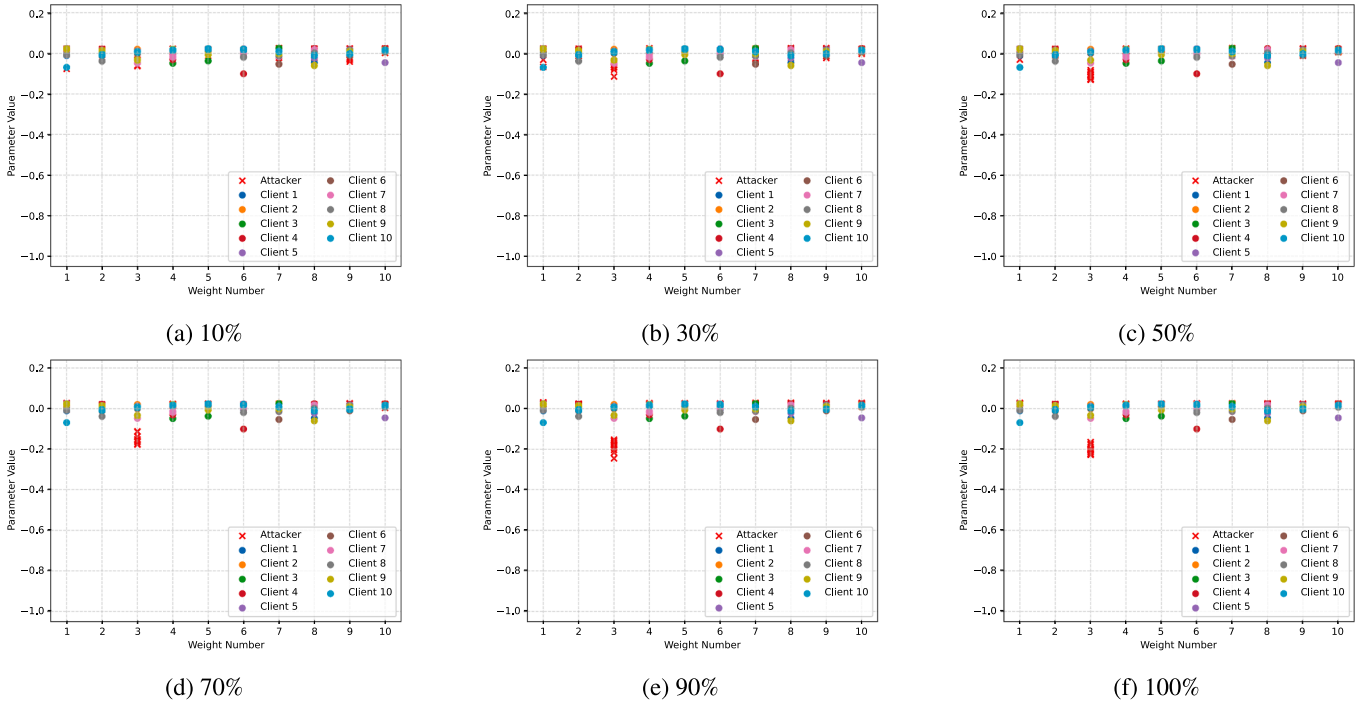
**Proportion of correctly recognized target labels (PCR).** PCR refers to the proportion of the number of iterations in which FLAB is able to correctly recognize the attacker-specified target label to the total number of iterations. The higher the PCR, the higher the accuracy of FLAB identifying malicious updates, the lower the BA.




(a) 10% (b) 30% (c) 50%

(d) 70% (e) 90% (f) 100%

**Fig. 4.** Distribution of biases in the output layer under different proportions of poisoned samples when the attacker conducts BadNets and the target label is 3 on MNIST.




(a) 10% (b) 30% (c) 50%

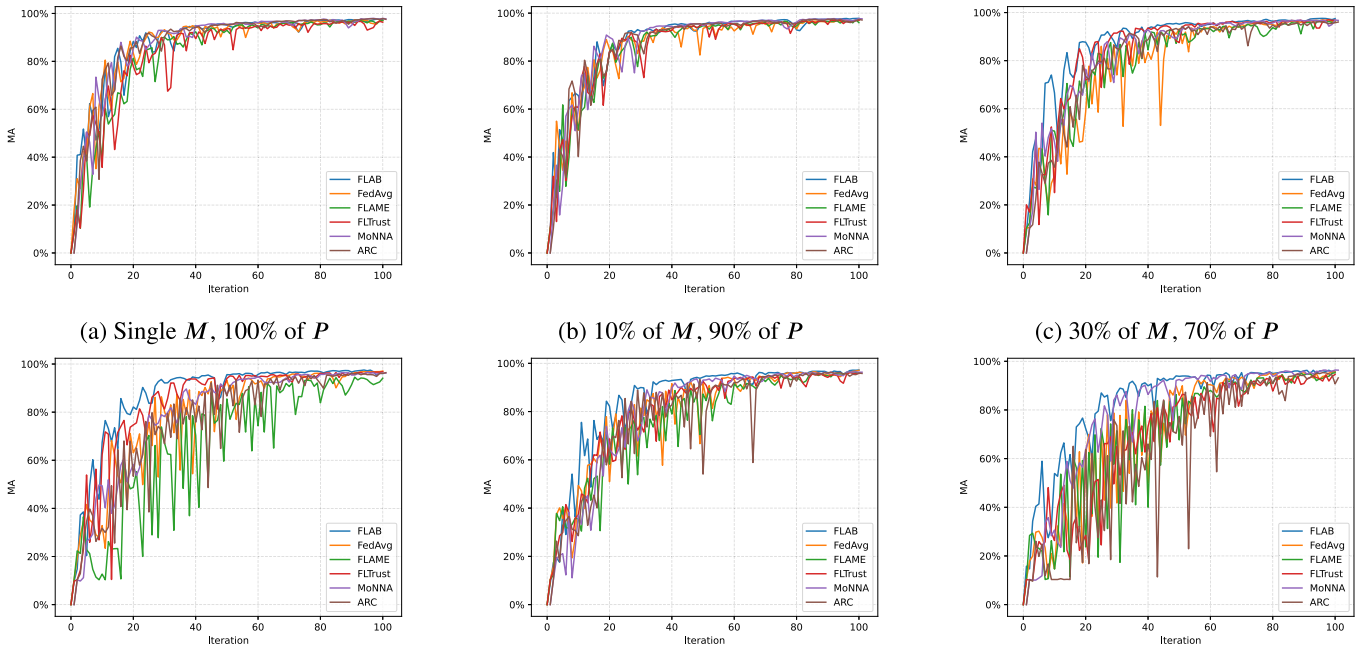
(d) 70% (e) 90% (f) 100%

**Fig. 5.** Distribution of weights in the output layer under different proportions of poisoned samples when the attacker conducts BadNets and the target label is 3 on MNIST.

### 6.2. Experimental results

#### 6.2.1. Distribution of biases and weights

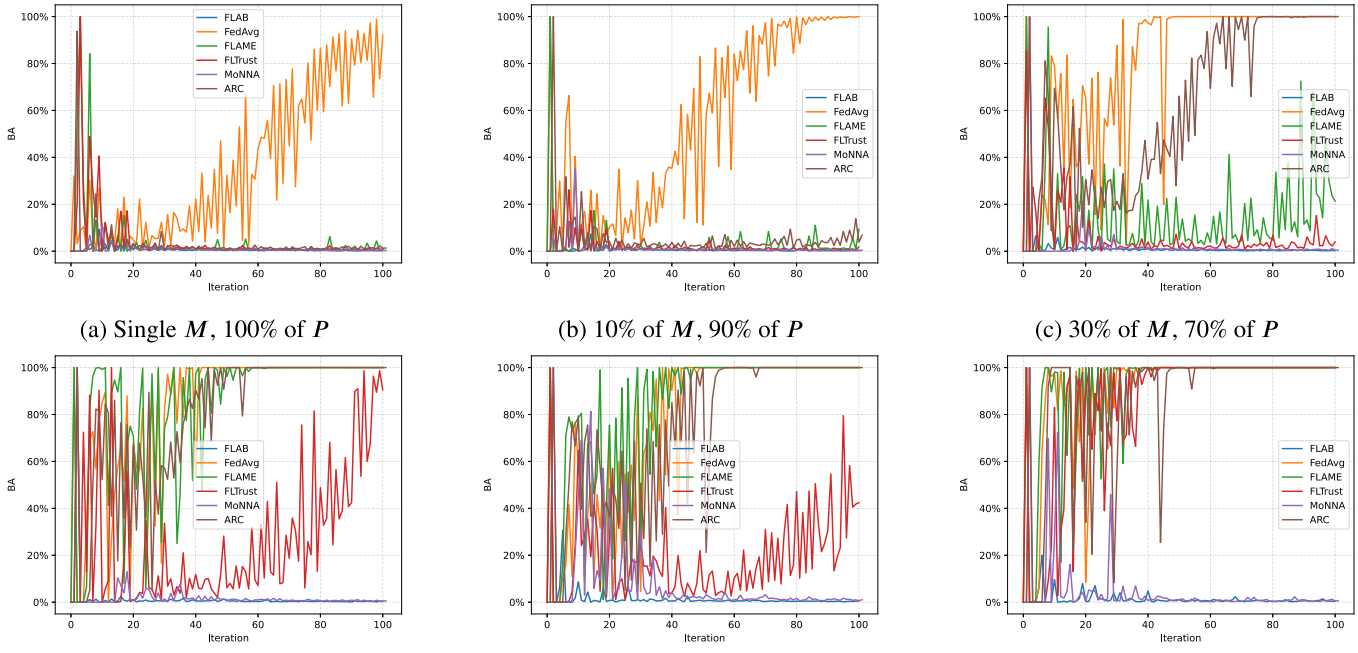
Figs. 4 and 5 illustrate the distribution of biases and weights with 50 % of compromised clients, where their parameters are uniformly labeled as "Attacker". When only 10 % of the samples are poisoned, all biases appear normal, resulting in the weakest attack impact. As the proportion of poisoned samples increases, the anomalous degree of biases labeled as 3 becomes progressively larger. When the proportion reaches 100 %, the anomalous degree reaches its maximum and all biases overlap. Moreover, even when the anomalous degree is not significantly enough (i.e., with 30 % of poisoned samples), FLAB can still effectively identify malicious updates. This is because FLAB considers both the number of anomaly biases and the anomalous degree of all




(a) Single $M$, 100% of $P$ (b) 10% of $M$, 90% of $P$ (c) 30% of $M$, 70% of $P$

(d) 50% of $M$, 50% of $P$ (e) 70% of $M$, 30% of $P$ (f) 90% of $M$, 40% of $P$

**Fig. 6.** MA of various algorithms in defending against DBA with different proportions of compromised clients ($M$) and poisoned samples ($P$) on MNIST.




(a) Single $M$, 100% of $P$ (b) 10% of $M$, 90% of $P$ (c) 30% of $M$, 70% of $P$

(d) 50% of $M$, 50% of $P$ (e) 70% of $M$, 30% of $P$ (f) 90% of $M$, 40% of $P$

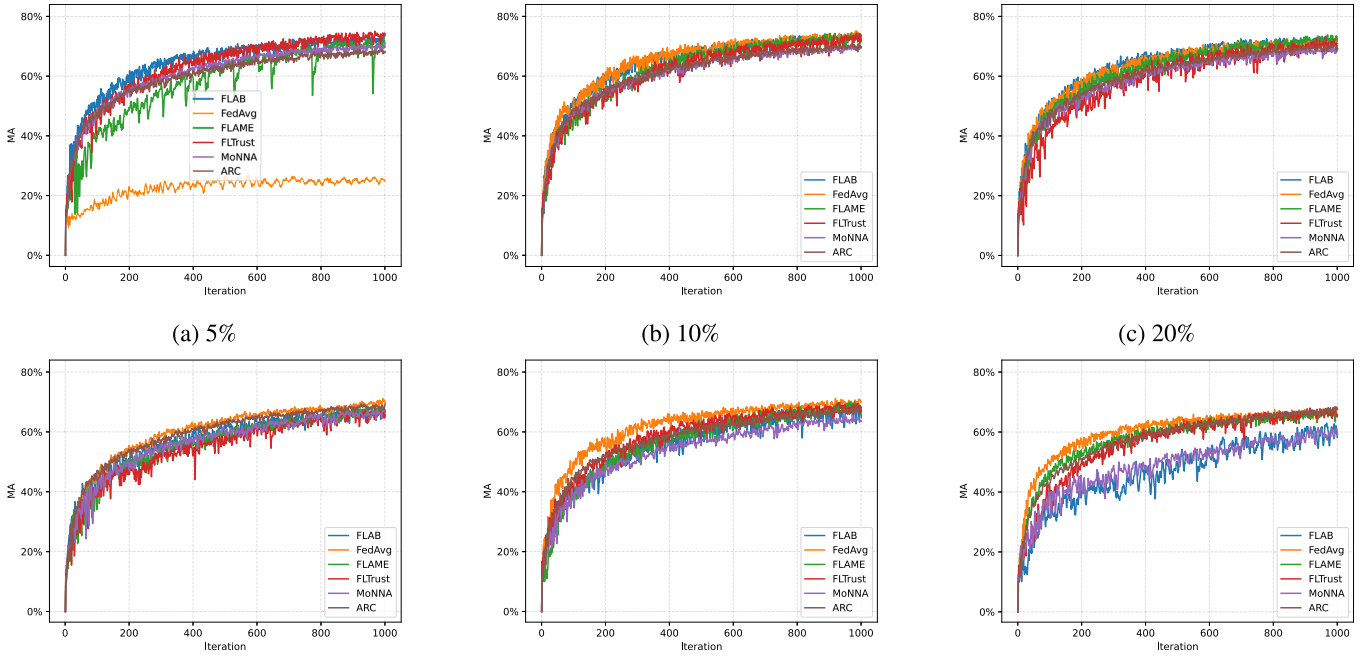
**Fig. 7.** BA of various algorithms in defending against DBA with different proportions of compromised clients ($M$) and poisoned samples ($P$) on MNIST.

biases. Compared with the biases, the anomalous degree of weights is not significant, so it is not considered in the defense.

Table 1 shows the performance of FLAB when using $w$ and $b$ under different proportion of poisoning samples. It can be clearly seen that compared with $w$, the use of $b$ allows the defense to identify the target label more accurately, and has a higher MA and a lower BA.

### 6.2.2. *Defend against backdoor attacks*

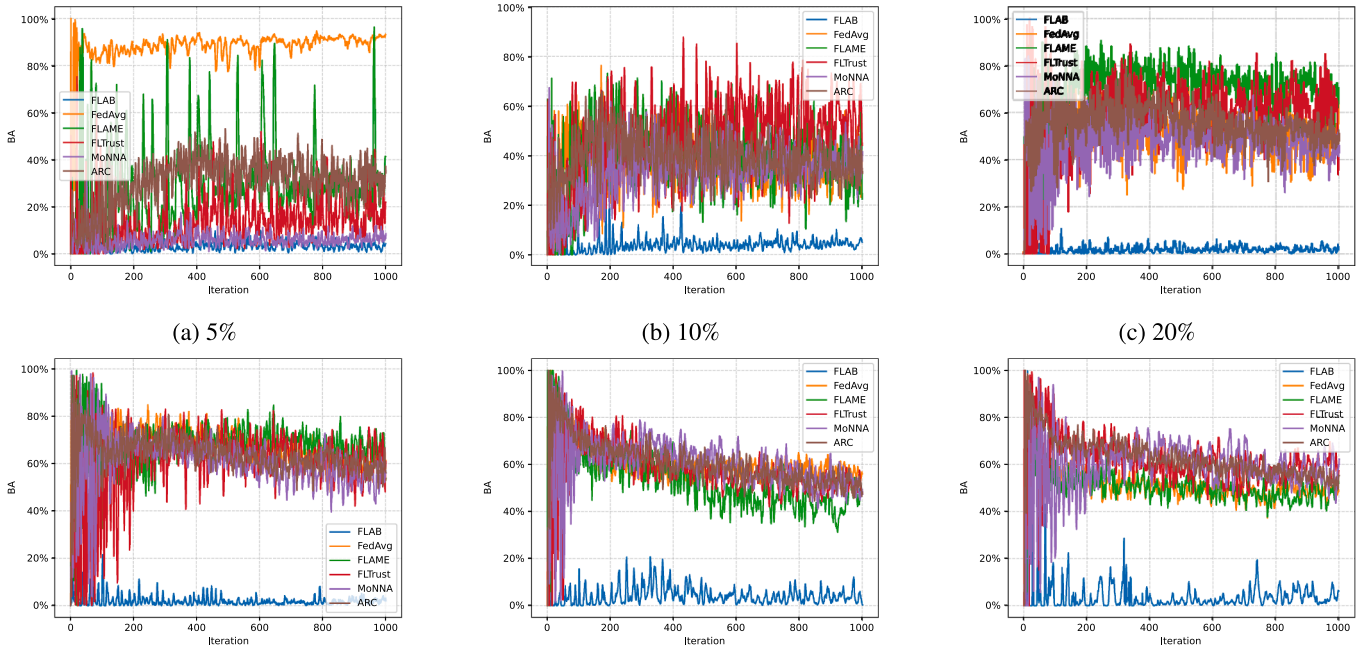
We compare BA and MA under different scenarios by varying the number of compromised clients and the proportion of poisoned samples. The number of compromised clients is set at 5 % (representing a single compromised client), 10 %, 30 %, 50 %, 70 %, and 90 %. Correspondingly, the proportion of poisoned samples is adjusted to 100 %, 90 %, 70 %, 50 %, 30 %, and 40 %. This setup ensures a meaningful evaluation




(a) 5% (b) 10% (c) 20%

(d) 40% (e) 60% (f) 80%

**Fig. 8.** MA of various algorithms in defending against HTB with different proportions of compromised clients and 42 % of poisoned samples on CIFAR10.




(a) 5% (b) 10% (c) 20%

(d) 40% (e) 60% (f) 80%

**Fig. 9.** BA of various algorithms in defending against HTB with different proportions of compromised clients and 42 % of poisoned samples on CIFAR10.

of the algorithms under varying attack intensities: (1) when the number of compromised clients is relatively small, a larger proportion of poisoned samples is needed to effectively implant a backdoor in the global model; (2) as the proportion of compromised clients increases, the proportion of poisoned samples is reduced to ensure that the MA remains effective and not excessively low. This approach provides a comprehensive assessment of each algorithm's robustness against different levels of backdoor attacks, balancing the trade-off between the success rates of backdoor attacks and maintaining model performance on the main task.

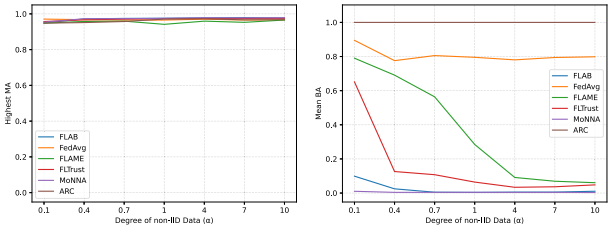
Figs. 6 and 7 illustrate the MA and BA of various algorithms in defending against DBA on MNIST, respectively. First, ARC effectively reduces BA only when the proportion of compromised clients is below 30 %. Both FLAME and FLTrust maintain relatively good MA, but their BA increases significantly once the proportion of compromised clients exceeds 30 % for FLAME and 50 % for FLTrust. Finally, FLAB

**Table 2**
Highest MA and mean BA of various algorithms under different proportions of compromised clients (M) and poisoned samples (P). The results are in the form of "Highest MA / mean BA (%)".

| Datasets | Attacks | Proportions (%) | | Defenses | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | M | P | FLAB | FedAvg | FLAME | FLTrust | MoNNA | ARC |
| MNIST | DBA | 5 | 100 | 97.64/0.22 | 97.28/85.99 | 97.08/1.35 | 96.70/0.89 | 97.96/0.44 | 97.86/1.33 |
| | | 10 | 90 | 97.83/0.24 | 97.07/99.70 | 97.22/2.25 | 97.15/0.81 | 97.72/0.42 | 97.51/4.63 |
| | | 30 | 70 | 97.61/0.32 | 97.02/80.62 | 96.31/14.85 | 97.04/4.50 | 96.93/0.61 | 96.26/98.65 |
| | | 50 | 50 | 97.52/0.33 | 97.08/84.92 | 94.54/86.04 | 96.83/28.32 | 96.55/0.69 | 96.41/100 |
| | | 70 | 30 | 97.23/0.36 | 97.09/77.81 | 96.53/84.62 | 96.19/19.70 | 96.49/1.17 | 96.30/100 |
| | | 90 | 40 | 96.33/0.67 | 95.69/91.68 | 95.28/91.82 | 94.62/84.63 | 96.42/0.79 | 95.35/100 |
| CIFAR10 | HTB | 5 | 42 | 74.31/3.17 | 26.98/88.24 | 72.28/28.72 | 75.06/13.64 | 71.29/7.01 | 68.80/30.48 |
| | | 10 | | 74.52/4.18 | 75.13/36.02 | 74.33/38.96 | 74.36/45.43 | 70/35.52 | 70.82/36.46 |
| | | 20 | | 73.35/1.97 | 72.71/52.90 | 73.82/69.29 | 73.29/57.52 | 69.92/43.85 | 70.66/50.18 |
| | | 40 | | 66.73/2.42 | 68.13/66.47 | 61.42/73.12 | 65.03/70.12 | 67.56/55.64 | 69.31/58.09 |
| | | 60 | | 69.28/4.16 | 71.20/60.78 | 70.11/53.54 | 69.82/57.92 | 65.50/49.26 | 68.34/50.89 |
| | | 80 | | 62.96/3.77 | 68.11/51.44 | 67.39/51.11 | 68.02/61.83 | 61.73/57.46 | 68.46/52.57 |

**Table 3**
Highest MA and mean BA of various algorithms in defending against different attacks on different datasets. The results are in the form of "Highest MA / mean BA (%)".

| Datasets | Attacks | Defenses | | | | | |
|---|---|---|---|---|---|---|---|
| | | FLAB | FedAvg | FLAME | FLTrust | MoNNA | ARC |
| MNIST | No attack | 97.58 | 98.03 | 97.56 | 97.73 | 97.73 | 97.67 |
| | BadNets | 97.47/0.60 | 96.53/83.28 | 94.01/87.18 | 96.60/6.65 | 97.03/0.42 | 87.16/100 |
| | LFA | 97.63 | 92.38 | 88.17 | 96.57 | 97.07 | 71.96 |
| CIFAR10 | No attack | 70.63 | 71.74 | 66.49 | 69.30 | 66.43 | 66.23 |
| | DBA | 67.36/0.24 | 66.08/89.78 | 60.23/92.33 | 59.19/70.90 | 56.06/95.02 | 56.64/99.56 |
| | BadNets | 67.84/2.63 | 67.06/93.87 | 64.33/95.68 | 64.93/93.50 | 56.71/97.57 | 58.38/97.90 |
| | LFA | 66.99 | 64.52 | 61.49 | 39.62 | 59.92 | 54.17 |
| Fashion-MNIST | No attack | 82.83 | 84.76 | 83.80 | 82.82 | 86.60 | 86.60 |
| | DBA | 82.11/2.71 | 79.86/67.15 | 74.41/68.87 | 80.51/13.85 | 86.24/3.30 | 10/100 |
| | BadNets | 84.42/1.47 | 81.30/79.61 | 78.33/82.00 | 79.95/12.21 | 85.92/2.90 | 10/100 |
| | LFA | 83.26 | 79.62 | 75.74 | 82.33 | 85.93 | 10 |




(a) Highest MA (b) Mean BA

**Fig. 10.** Highest MA and mean BA of various algorithms in defending against BadNets on MNIST with different $\alpha$.

and MoNNA achieve the best performance, with MA that is comparable to or better than the other algorithms, and BA close to zero across all settings. Therefore, it can be concluded that FLAB demonstrates excellent performance in defending against DBA.

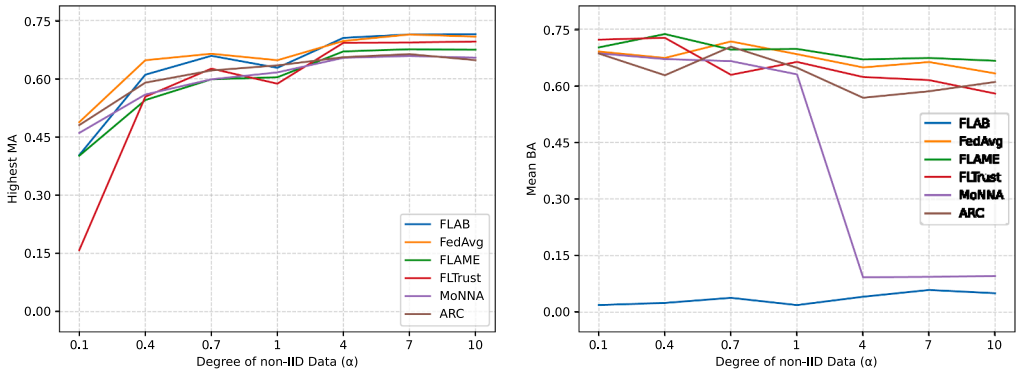
Figs. 8 and 9 depict MA and BA of various algorithms in defending against HTB on CIFAR10, respectively. In this scenario, we use a fixed proportion of poisoned samples, i.e., 42 %, and vary the proportion of compromised clients, i.e., 5 %, 10 %, 20 %, 40 %, 60 %, and 80 %. Overall, other robust aggregation schemes, except FLAB, rely on traditional anomaly detection mechanisms and do not fully leverage the characteristics of backdoor attackers. As a result, their BA performance is generally average and noticeably inferior to that of FLAB. FLAB's MA shows a slight decrease when the proportion compromised clients reaches 80 %. The reduction occurs because FLAB aggressively removes malicious updates instead of mildly reducing their impact, resulting in an insufficient number of benign model updates and a less optimal fit of the global model to the non-IID data of all clients. However, this approach allows FLAB to significantly improve BA at the cost of a small reduction in MA. The results of the above experiments are specifically summarized in Table 2.

Table 3 provides additional evaluation of various algorithms in defending against different attacks across multiple datasets, with the proportion of compromised clients set to 50 %. ARC consistently performs the worst and demonstrates no clear advantage on any dataset. Both MoNNA and FLTrust generally perform well; however, their BA increases significantly as the dataset complexity rises, particularly in the case of CIFAR-10. FLAME performs poorly across all settings due to the failure of its median-based mechanism when the proportion of compromised clients reaches or exceeds 50 %. In contrast, FLAB outperforms all other methods, achieving higher MA and lower BA across different attacks and datasets, thereby demonstrating strong Byzantine robustness.

### 6.2.3. *Impact of data distributions*

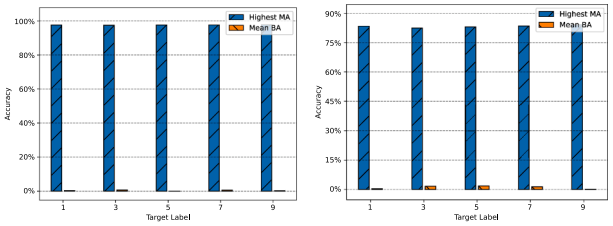
As depicted in Figs. 10 and 11, we evaluate highest MA and mean BA of various algorithms under different data distributions. The parameter $\alpha$ indicates the degree of non-IID data, where a smaller $\alpha$ value signify a higher non-IID degree; $\alpha = 10$ suggests nearly IID client data. To ensure all algorithms function properly, we limit the proportion of compromised clients to less than 50 %, e.g., 40 %. For relatively straightforward MNIST, varying $\alpha$ values have no significant impact on the highest MA across all robust algorithms. Conversely, CIFAR10, being more complex, shows a more noticeable impact of $\alpha$ on the highest MA, generally deteriorating as $\alpha$ decreases.

The mean BA is similarly influenced by the value of $\alpha$. On MNIST, ARC provides no defense against BadNets. Although FLAME and FLTrust offer moderate protection, their performance deteriorates as $\alpha$ decreases. In contrast, FLAB and MoNNA achieve the best results,



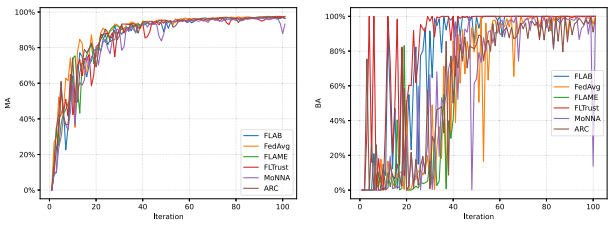

(a) Highest MA (b) Mean BA

**Fig. 11.** Highest MA and mean BA of various algorithms in defending against HTB on CIFAR10 with different $\alpha$.



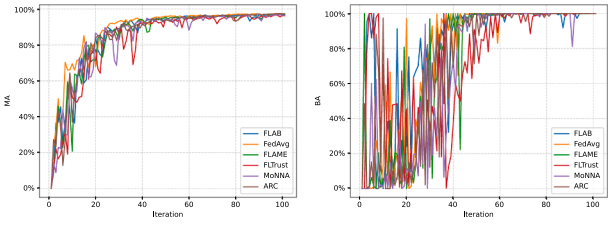

(a) (b)

**Fig. 12.** Highest MA and mean BA of FLAB under different target labels assigned by the attacker.




(a) Highest MA (b) Mean BA

**Fig. 13.** Highest MA and mean BA of various algorithms in defending against clean-label backdoor attacks.




(a) Highest MA (b) Mean BA

**Fig. 14.** Highest MA and mean BA of various algorithms when the attacker marks samples label 7 as 3.

consistently maintaining a very low mean BA. On CIFAR-10, the performance of most robust algorithms-including FLAME, FLTrust, and ARC-becomes unsatisfactory. Although MoNNA performs remarkably well overall, its mean BA is highly sensitive to $\alpha$; specifically, the attack success rate increases significantly when $\alpha < 1$. In comparison, FLAB continues to perform well and clearly outperforms all other robust algorithms.

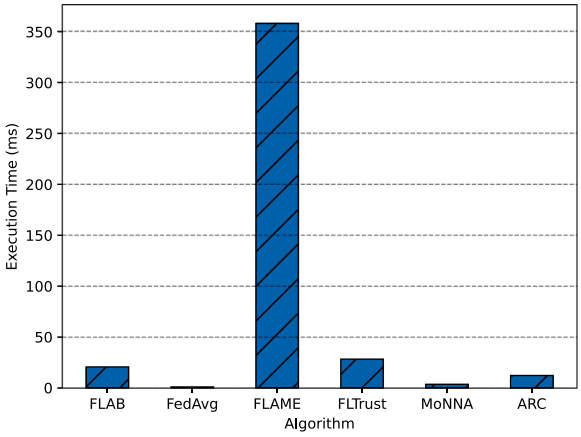



**Fig. 15.** Execution time of different robust algorithms when defending against DBA on FashionMNIST.

In summary, FLAB demonstrates effectiveness across diverse data distributions, including scenarios with extreme non-IID and IID data.

### 6.2.4. *Impact of different target labels*

To investigate FLAB's resistance to varying target labels, we conduct experiments to assess the impact of different target labels on FLAB's highest MA and mean BA across the default number of iterations, as depicted in Fig. 12. Fig. 12a and b illustrate FLAB defending against DBA on MNIST and against BadNets on Fashion-MNIST, respectively. Our results clearly demonstrate that changing the target label has minimal effect on FLAB's performance. FLAB consistently proves effective in defending against backdoor attacks across diverse settings, underscoring its robustness and reliability against different target labels.

### 6.2.5. *Impact of label distribution*

In previous experiments, the attacker marks a specified percentage of arbitrarily labeled samples as the target label for backdoor attacks. In Figs. 13 and 14, we analyze the performance of various algorithms without changing the distribution of labels, i.e., the attacker marks samples with label 3 as 3 and samples with label 7 as 3, respectively. Unfortunately, FLAB does not effectively reduce the success rate of backdoor attacks, so it is not applicable to this attack scenario.

Intuitively, FLAB identifies the attacker's target label by analyzing the label distribution across all clients. For instance, if the attacker's target label is 3, this label tends to appear with a higher frequency in the label distributions of compromised clients. However, this assumption breaks down in backdoor attacks where the overall label distribution

remains unchanged. In such attacks, the attacker manipulates compromised clients' samples with a certain label rather than changing their label distributions. Since clients' data is non-IID, the proportion of labels for these samples varies across compromised clients, making it difficult for FLAB to detect the anomaly bias associated with the target label. Given that FLAB is fundamentally designed to leverage label distributions, it offers limited effectiveness in defending against such attacks.

In backdoor attacks involving changed label distributions, the attacker can compromise any number of clients and inject an arbitrary quantity of backdoor samples. Consequently, such attacks often achieve higher success rates and exhibit greater destructive impact compared to those that leave label distributions unchanged. In these scenarios, FLAB demonstrates a clear advantage, as it can effectively mitigate the attack success rate without relying on prior assumptions. Moreover, in real-world FL systems with high security demands, the strengths of FLAB become particularly pronounced.

#### 6.2.6. Comparison of runtime efficiency

Fig. 15 illustrates the execution time of different algorithms per iteration when defending against DBA, assuming 50 % of clients are malicious, each with 50 % of their data consisting of backdoor samples. Among these algorithms, FLAME shows relatively low runtime efficiency due to the computational overhead of HDBSCAN clustering. In contrast, FedAvg performs only simple averaging of model updates, resulting in the highest efficiency. Compared to other robust algorithms, FLAB achieves relatively high efficiency. More importantly, it consistently maintains a significantly lower BA, giving it a clear overall advantage in both robustness and efficiency.

## 7. Discussion

Beyond its empirical effectiveness, FLAB is designed with practical applicability in mind. First, it operates entirely on the server side without requiring any modifications to the client training process, ensuring seamless compatibility with existing FL frameworks. Second, it does not rely on prior knowledge of the number of attackers, which aligns with real-world deployment constraints. Finally, FLAB remains effective under non-IID data distributions and partial client participation. These properties make FLAB not only theoretically robust but also readily deployable in practical FL systems, highlighting its potential contribution to enhancing the security and reliability of distributed machine learning systems.

In the experimental setup of Section 6, we assume that over 50 % of clients may be compromised. This represents a worst-case scenario, as the proportion of compromised clients in real-world environments is expected to be much lower. Such a setup allows us to evaluate the robustness of the proposed method under severe adversarial conditions. Notably, our approach also performs well when the compromised client ratio is below 50 %, a scenario more representative of practical deployments. Moreover, our dataset partitioning strategy employs a Dirichlet distribution, without explicitly modeling edge device heterogeneity or network constraints. This choice aligns with standard practice in Byzantine-robust FL research, which primarily evaluates attack resistance under controlled conditions rather than focusing on system-level factors such as computational heterogeneity. These simplifications do not significantly affect our experimental results or conclusions. Nonetheless, we acknowledge the value of validation under real-world conditions. In future work, we plan to extend our evaluation to more realistic scenarios, incorporating asynchronous aggregation, heterogeneous hardware simulation, and variable communication delays, thereby assessing the algorithm in authentic heterogeneous device environments.

Although FLAB is effective in reducing the success rate of backdoor attacks, it does not provide a robust defense when the attacker targets only samples of a specific label, i.e., without changing the label distributions. In future work, we plan to investigate the characteristics of such attacks in order to develop a more comprehensive robust aggregation algorithm.

## 8. Conclusions

In this paper, we propose FLAB, a novel defense mechanism designed to defend against backdoor attacks in Federated learning (FL). Unlike conventional anomaly detection approaches, FLAB utilizes characteristics such as the target and intensity of backdoor attacks to detect malicious model updates. Specifically, we introduce anomaly bias to characterize updates and design a detection mechanism to quantify these biases. Furthermore, we determine the attacker's target label by clustering anomaly biases and gradually reducing cluster sizes. Experimental results showcase FLAB's ability to achieve exceptionally low backdoor accuracy without the need for auxiliary knowledge. Importantly, FLAB maintains its effectiveness even when a significant proportion of clients are compromised.

## CRediT authorship contribution statement

**Hua Wang:** Conceptualization,Supervision,Formal analysis,Writing – review & editing; **Shaoxiong Wang:** Conceptualization,Methodology,Software,Writing – original draft,Writing – review & editing; **Lianhua Wang:** Investigation,Resources,Formal analysis,Data curation; **Rui Wang:** Validation,Visualization,Investigation.

## 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.

## Acknowledgments

This research is partially funded by the Foundation for Quality Professional Degree Teaching Case Library for Graduate Students in Shandong Province(No. SDYAL2024165) and Shandong Provincial Natural Science Foundation (ZR2024MF146). The authors are indebted to their support.

## Appendix A.

**Proof of theorem**. First, We start from the update rule

$$G^{t+1} = G^t - \alpha \hat{g}^t,$$

where $\hat{g}^t$ is the average gradient of clients selected after anomaly detection, which may have a certain deviation.

To analyze its relationship with the optimal solution $G^*$, we consider the distance change between the model and the optimal solution

$$\begin{aligned}
&\mathbb{E}[\|G^{t+1} - G^*\|^2] \\
&= \mathbb{E}[\|G^t - G^* - \alpha \hat{g}^t\|^2] \\
&= \mathbb{E}[\|G^t - G^*\|^2] - 2\alpha \mathbb{E}[\langle G^t - G^*, \hat{g}^t \rangle] + \alpha^2 \mathbb{E}[\|\hat{g}^t\|^2].
\end{aligned}$$

Since $\hat{g}^t$ may be biased, we represent the deviation between the current aggregate gradient and the real global gradient as

$$e_t := \mathbb{E}[\hat{g}^t] - \nabla F(G^t), \quad \|e_t\| \le b_t.$$

Then we can decompose the inner product into two parts

$$\begin{aligned}
&\mathbb{E}[\langle G^t - G^*, \hat{g}^t \rangle] \\
&= \mathbb{E}[\langle G^t - G^*, \nabla F(G^t) \rangle] + \mathbb{E}[\langle G^t - G^*, e_t \rangle].
\end{aligned}$$

By Assumption 1 and Cauchy-Schwarz inequality, we can get

$$\langle G^t - G^*, \nabla F(G^t)\rangle \geq \mu\|G^t - G^*\|^2,$$

$$|\langle G^t - G^*, e_t\rangle| \leq \|G^t - G^*\| \cdot \|e_t\| \leq b_t\|G^t - G^*\|.$$

Comprehensively, the lower boundary can be obtained

$$\mathbb{E}[\langle G^t - G^*, \hat{g}^t\rangle] \geq \mu\mathbb{E}[\|G^t - G^*\|^2] - b_t\mathbb{E}[\|G^t - G^*\|].$$

Next, we use the second-order moment to decompose

$$\mathbb{E}[\|\hat{g}^t\|^2] = \|\mathbb{E}[\hat{g}^t]\|^2 + \mathrm{Var}(\hat{g}^t).$$

By Assumptions 1 and 2, we can get

$$\begin{aligned}\|\mathbb{E}[\hat{g}^t]\|^2 &\leq (\|\nabla F(G^t)\| + \|e_t\|)^2 \\ &\leq (L\|G^t - G^*\| + b_t)^2 \\ &\leq 2L^2\|G^t - G^*\|^2 + 2b_t^2,\end{aligned}$$

$$\mathrm{Var}(\hat{g}^t) \leq \frac{\sigma^2}{|S_t|} \leq \sigma^2.$$

To sum up

$$\mathbb{E}[\|\hat{g}^t\|^2] \leq 2L^2\mathbb{E}[\|G^t - G^*\|^2] + 2b_t^2 + \sigma^2.$$

Therefore, we can get the convergence boundary

$$\begin{aligned}&\mathbb{E}[\|G^{t+1} - G^*\|^2] \\ &\leq \mathbb{E}[\|G^t - G^*\|^2] \\ &\quad -2\alpha\left(\mu\mathbb{E}[\|G^t - G^*\|^2] - b_t\mathbb{E}[\|G^t - G^*\|]\right) \\ &\quad +\alpha^2\left(2L^2\mathbb{E}[\|G^t - G^*\|^2] + 2b_t^2 + \sigma^2\right) \\ &\leq \left(1 - 2\alpha\mu + 2\alpha^2L^2\right)\mathbb{E}[\|G^t - G^*\|^2] \\ &\quad +\alpha^2(2b_t^2 + \sigma^2) + 2\alpha b_t\mathbb{E}[\|G^t - G^*\|].\end{aligned}$$

This completes the proof.

## References


Allouah, Y., Guerraoui, R., Gupta, N., Jellouli, A., Rizk, G., & Stephan, J. (2024). The vital role of gradient clipping in byzantine-resilient distributed learning. *arXiv preprint arXiv:2405.14432*.

Andreina, S., Marson, G. A., Möllering, H., & Karame, G. (2021). Baffle: Backdoor detection via feedback-based federated learning. In *2021 IEEE 41st international conference on distributed computing systems (ICDCS)* (pp. 852–863). IEEE.

Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In *International conference on artificial intelligence and statistics* (pp. 2938–2948). PMLR.

Baruch, G., Baruch, M., & Goldberg, Y. (2019). A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, *32* (pp. 8632–8642).

Bhagoji, A. N., Chakraborty, S., Mittal, P., & Calo, S. (2019). Analyzing federated learning through an adversarial lens. In *International conference on machine learning* (pp. 634–643). PMLR.

Biggio, B., Nelson, B., & Laskov, P. (2012). Poisoning attacks against support vector machines. *arXiv preprint arXiv:1206.6389*.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, *30* (pp. 119–129).

Campello, R. J., Moulavi, D., & Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 160–172). Springer.

Cao, X., Fang, M., Liu, J., & Gong, N. Z. (2020). Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*.

Cao, X., & Gong, N. Z. (2022). MPAF: Model poisoning attacks to federated learning based on fake clients. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3396–3404).

Chen, X., Liu, C., Li, B., Lu, K., & Song, D. (2017a). Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Chen, Y., Su, L., & Xu, J. (2017b). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, *1*(2), 1–25.

Chu, T., Garcia-Recuero, A., Iordanou, C., Smaragdakis, G., & Laoutaris, N. (2022). Securing federated sensitive topic classification against poisoning attacks. *arXiv preprint arXiv:2201.13086*.

Ding, B., Yang, P., & Huang, S.-J. (2025). FedDLAD: A federated learning dual-layer anomaly detection framework for enhancing resilience against backdoor attacks. In J. Kwok (Ed.), *Proceedings of the thirty-fourth international joint conference on artificial intelligence, IJCAI-25* (pp. 5021–5029). International Joint Conferences on Artificial Intelligence Organization. Main Track https://doi.org/10.24963/ijcai.2025/559

Fang, M., Cao, X., Jia, J., & Gong, N. (2020). Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX security 20)* (pp. 1605–1622).

Farhadkhani, S., Guerraoui, R., Gupta, N., Hoang, L.-N., Pinot, R., & Stephan, J. (2023). Robust collaborative learning with linear gradient overhead. In *International conference on machine learning* (pp. 9761–9813). PMLR.

Fung, C., Yoon, C. J. M., & Beschastnikh, I. (2020). The limitations of federated learning in sybil settings. In *23rd international symposium on research in attacks, intrusions and defenses (RAID 2020)* (pp. 301–316).

Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

El Mhamdi, E. M., Guerraoui, R., & Rouault, S. (2018). The hidden vulnerability of distributed learning in byzantium. In *International conference on machine learning* (pp. 3521–3530). PMLR.

Han, X., Xu, G., Zhou, Y., Yang, X., Li, J., & Zhang, T. (2022). Physical backdoor attacks to lane detection systems in autonomous driving. In *Proceedings of the 30th ACM International conference on multimedia* (pp. 2957–2968).

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., & Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Kim, T., Oh, J., Kim, N., Cho, S., & Yun, S.-Y. (2021). Comparing Kullback-Leibler divergence and mean squared error loss in knowledge distillation. *arXiv preprint arXiv:2105.08919*.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krauß, T., & Dmitrienko, A. (2023). Mesas: Poisoning defense for federated learning resilient against adaptive attackers. In *Proceedings of the 2023 ACM SIGSAC conference on computer and communications security* (pp. 1526–1540).

Krizhevsky, A., Hinton, G. et al. (2009). Learning multiple layers of features from tiny images. Technical Report, Department of Computer Science, University of Toronto. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J. Z., Langer, D., Pink, O., Pratt, V., Sokolsky, M., Stanek, G., Stavens, D. M., Teichman, A., Werling, M., & Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent vehicles symposium (IV)* (pp. 163–168). IEEE.

Li, S., Cheng, Y., Wang, W., Liu, Y., & Chen, T. (2020). Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, *1*(2) (pp. 209–230).

Liu, J., Peng, C., Tan, W., & Shi, C. (2024). Federated learning backdoor attack based on frequency domain injection. *Entropy*, *26*(2), 164.

Long, G., Tan, Y., Jiang, J., & Zhang, C. (2020). Federated learning for open banking. In *Federated learning: Privacy and incentive* (pp. 240–254). Springer.

Lv, Z., & Song, H. (2019). Mobile internet of things under data physical fusion technology. *IEEE Internet of Things Journal*, *7*(5), 4616–4624.

Mao, A., Mohri, M., & Zhong, Y. (2023). Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on machine learning* (pp. 23803–23828). PMLR.

McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics* (pp. 1273–1282). PMLR.

Muñoz-González, L., Co, K. T., & Lupu, E. C. (2019). Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*.

Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021a). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1622–1658.

Nguyen, T. D., Rieger, P., Chen, H., Yalame, H., Möllering, H., Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Zeitouni, S., Koushanfar, F., Sadeghi, A.-R., & Schneider, T. (2022). {FLAME}: Taming backdoors in federated learning. In *31st USENIX security symposium (USENIX security 22)* (pp. 1415–1432).

Nguyen, T. D., Rieger, P., Miettinen, M., & Sadeghi, A.-R. (2020). Poisoning attacks on federated learning-based iot intrusion detection system. In *Proc. workshop decentralized iot syst. secur.(DISS). (vol. 79)*.

Nguyen, T. D., Rieger, P., Yalame, H., Möllering, H., Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Sadeghi, A.-R., Schneider, T., & Zeitouni, S. (2021b). Flguard: Secure and private federated learning. *IACR Cryptol. ePrint Arch.*, *2021*, 25.

Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K. H., Ourselin, S., Sheller, M. J., Summers, R. M., Trask, A., Xu, D., Baust, M., & Cardoso, M. J. (2020). The future of digital health with federated learning. *NPJ Digital Medicine*, *3*(1), 1–7.

Rubinstein, B. I. P., Nelson, B., Huang, L., Joseph, A. D., Lau, S.-h., Rao, S., Taft, N., & Tygar, J. D. (2009). AntiDote: Understanding and defending against poisoning of anomaly detectors. In *Proceedings of the 9th ACM SIGCOMM conference on internet measurement* (pp. 1–14).

Sattler, F., Müller, K.-R., & Samek, W. (2020a). Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(8), 3710–3722.

Sattler, F., Müller, K.-R., Wiegand, T., & Samek, W. (2020b). On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International conference on acoustics, speech and signal processing (icassp)* (pp. 8861–8865). IEEE.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., & Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, *31* (pp. 6106–6116).

Shejwalkar, V., & Houmansadr, A. (2021). Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

Shen, S., Tople, S., & Saxena, P. (2016). Auror: Defending against poisoning attacks in collaborative deep learning systems. In *Proceedings of the 32nd annual conference on computer security applications* (pp. 508–519).

Steinhardt, J., Koh, P. W. W., & Liang, P. S. (2017). Certified defenses for data poisoning attacks. *Advances in Neural Information Processing Systems*, *30* (pp. 3517-3529).

Sun, Z., Kairouz, P., Suresh, A. T., & McMahan, H. B. (2019). Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*.

Wan, W., Hu, S., Li, M., Lu, J., Zhang, L., Zhang, L. Y., & Jin, H. (2023). A four-pronged defense against byzantine attacks in federated learning. In *Proceedings of the 31st ACM International conference on multimedia* (pp. 7394–7402).

Wang, H., Sreenivasan, K., Rajput, S., Vishwakarma, H., Agarwal, S., Sohn, J.-y., Lee, K., & Papailiopoulos, D. (2020). Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, *33*, 16070–16084.

Wu, C., Yang, X., Zhu, S., & Mitra, P. (2020). Mitigating backdoor attacks in federated learning. *arXiv preprint arXiv:2011.01767*.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Xie, C., Huang, K., Chen, P.-Y., & Li, B. (2019). DBA: Distributed backdoor attacks against federated learning. In *International conference on learning representations*.

Yang, H., Zhang, X., Fang, M., & Liu, J. (2019). Byzantine-resilient stochastic gradient descent for distributed learning: A lipschitz-inspired coordinate-wise median approach. In *2019 IEEE 58th conference on decision and control (CDC)* (pp. 5832–5837). IEEE.

Yin, D., Chen, Y., Kannan, R., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning* (pp. 5650–5659). Pmlr.

Yuan, J., Zhang, Q., Chen, N., Chen, S., & Xu, B. (2025). A multi-granularity clustering approach for federated backdoor defense with the adam optimizer. In J. Kwok (Ed.), *Proceedings of the thirty-fourth International joint conference on artificial intelligence, IJCAI-25* (pp. 6931–6939). International Joint Conferences on Artificial Intelligence Organization. Main Track https://doi.org/10.24963/ijcai.2025/771

Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. *Knowledge-Based Systems*, *216*, 106775.

Zhang, Q., Yu, M., Wang, R., Li, Y., Yuan, J., & Tan, Y.-a. (2025). A sparse and invisible targeted backdoor attack in federated learning. *Journal of King Saud University Computer and Information Sciences*, *37*(6), 1–13.

Zhang, Z., Cao, X., Jia, J., & Gong, N. Z. (2022). 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* (pp. 2545–2555).