FLAME: Taming Backdoors in Federated Learning

Thien Duc Nguyen and Phillip Rieger, *Technical University of Darmstadt;* Huili Chen, *University of California San Diego;* Hossein Yalame, Helen Möllering, and Hossein Fereidooni, *Technical University of Darmstadt;* Samuel Marchal, *Aalto University and F-Secure;* Markus Miettinen, *Technical University of Darmstadt;* Azalia Mirhoseini, *Google;* Shaza Zeitouni, *Technical University of Darmstadt;* Farinaz Koushanfar, *University of California San Diego;* Ahmad-Reza Sadeghi and Thomas Schneider, *Technical University of Darmstadt*

https://www.usenix.org/conference/usenixsecurity22/presentation/nguyen

This paper is included in the Proceedings of the
31st USENIX Security Symposium.

August 10–12, 2022 • Boston, MA, USA

978-1-939133-31-1

FLAME: Taming Backdoors in Federated Learning

Thien Duc Nguyen[1], Phillip Rieger[1], Huili Chen[2], Hossein Yalame[1], Helen Möllering[1],
Hossein Fereidooni[1], Samuel Marchal[3], Markus Miettinen[1], Azalia Mirhoseini[4],
Shaza Zeitouni[1], Farinaz Koushanfar[2], Ahmad-Reza Sadeghi[1], and Thomas Schneider [1]
[1]*Technical University of Darmstadt, Germany*;* [2]*University of California San Diego, USA*; [3]*Aalto University and F-Secure, Finland*; [4]*Google, USA*

Abstract

Federated Learning (FL) is a collaborative machine learning approach allowing participants to jointly train a model without having to share their private, potentially sensitive local datasets with others. Despite its benefits, FL is vulnerable to so-called *backdoor attacks*, in which an adversary injects manipulated model updates into the federated model aggregation process so that the resulting model will provide targeted false predictions for specific adversary-chosen inputs. Proposed defenses against backdoor attacks based on detecting and filtering out malicious model updates consider only very specific and limited attacker models, whereas defenses based on differential privacy-inspired noise injection significantly deteriorate the benign performance of the aggregated model. To address these deficiencies, we introduce FLAME, a defense framework that estimates the sufficient amount of noise to be injected to ensure the elimination of backdoors. To minimize the required amount of noise, FLAME uses a model clustering and weight clipping approach. This ensures that FLAME can maintain the benign performance of the aggregated model while effectively eliminating adversarial backdoors. Our evaluation of FLAME on several datasets stemming from application areas including image classification, word prediction, and IoT intrusion detection demonstrates that FLAME removes backdoors effectively with a negligible impact on the benign performance of the models.

1 Introduction

Federated learning (FL) is an emerging collaborative machine learning trend with many applications, such as next word prediction for mobile keyboards [39], medical imaging [49], and intrusion detection for IoT [44] to name a few. In FL, clients locally train models based on local training data and then provide these model updates to a central aggregator who combines them into a global model. The global model is then propagated back to the clients for the next training iteration.

FL promises efficiency and scalability as the training is distributed among many clients and executed in parallel. In particular, FL improves privacy by enabling clients to keep their training data locally [38]. Despite its benefits, FL has been shown to be vulnerable to so-called *poisoning attacks* where the adversary manipulates the local models of a subset of clients participating in the federation so that the malicious updates get aggregated into the global model. Untargeted poisoning attacks merely aim at deteriorating the performance of the global model and can be defeated by validating the performance of uploaded models [12]. In this paper, we therefore focus on the more challenging problem of *backdoor attacks* [7, 45, 57, 59], i.e., targeted poisoning attacks in which the adversary seeks to stealthily manipulate the resulting global model in a way that attacker-controlled inputs result in incorrect predictions chosen by the adversary.

Deficiencies of existing defenses. Existing defenses against backdoor attacks can be roughly divided into two categories: The first one comprises anomaly detection-based approaches [4,9,22,51] for identifying and removing potentially poisoned model updates. However, these solutions are effective only under very specific adversary models, as they make detailed assumptions about the attack strategy of the adversary and/or the underlying distribution of the benign or adversarial datasets. If these very specific assumptions do not hold, the defenses may fail. The second category is inspired by differential privacy (DP) techniques [7,56], where individual weights[1] of model updates are clipped to a maximum threshold and random noise is added to the weights for diluting/reducing the impact of potentially poisoned model updates on the aggregated global model. In contrast to the first category, DP techniques [7,56] are applicable in a generic adversary model without specific assumptions about adversarial behavior and data distributions and are effective in eliminating the impact of malicious model updates. However, straightforward application of DP approaches severely deteriorates the benign

*Emails: {ducthien.nguyen, ahmad.sadeghi}@trust.tu-darmstadt.de

[1] Parameters of neural network models typically consist of 'weights' and 'biases'. For the purposes of this paper, however, these parameters can be treated identically and we will refer to them as 'weights' for brevity.

performance of the aggregated model because the amount of noise required to ensure effective elimination of backdoors also results in significant modifications of individual weights of benign model updates [7,57].

In this paper, we develop a resilient defense against backdoors by combining the benefits of both defense types without suffering from the limitations (narrow attacker model, assumptions about data distributions) and drawbacks (loss of benign performance) of existing approaches. To this end, we introduce an approach in which detection of anomalous model updates and tuned clipping of weights are combined to minimize the amount of noise needed for backdoor removal of the aggregated model while preserving its benign performance.

Our Goals and Contributions. We present FLAME, a resilient aggregation framework for FL that eliminates the impact of backdoor attacks while maintaining the benign performance of the aggregated model. This is achieved by three modules: DP-based noising of model updates to remove backdoor contributions, automated model clustering approach to identify and eliminate potentially poisoned model updates, and model weight clipping before aggregation to limit the impact of malicious model updates on the aggregation result. The last two modules can significantly reduce the amount of random noise required by DP noising for backdoor elimination. In particular, our contributions are as follows:

- We present FLAME, a defense framework against backdoor attacks in FL that is capable of eliminating backdoors without impacting the benign performance of the aggregated model. Contrary to earlier backdoor defenses, FLAME is applicable in a *generic* adversary model, i.e., it does not rely on strong assumptions about the attack strategy of the adversary, nor about the underlying data distributions of benign and adversarial datasets (§4.1).
- We show that the amount of required Gaussian noise can be radically reduced by: a) applying our clustering approach to remove potentially malicious model updates and b) clipping the weights of local models at a proper level to constrain the impact of individual (especially malicious) models on the aggregated model. (§4.3)
- We provide a noise boundary proof for the amount of Gaussian noise required by noise injection (inspired by DP) to eliminate backdoor contributions (§5).
- We extensively evaluate our defense framework on real-world datasets from three very different application areas. We show that FLAME reduces the amount of required noise so that the benign performance of the aggregated model does not degrade significantly, providing a crucial advantage over state-of-the-art defenses using straightforward injection of DP-based noise (§7).

As an orthogonal aspect, we also consider how the privacy of model updates against an honest-but-curious aggregator can be preserved and develop a secure multi-party computation approach that can preserve the privacy of individual model updates while realizing our backdoor defense approach (§8).

2 Background and Problem Setting

2.1 Federated Learning

Federated Learning [38,50] is a concept for distributed machine learning that links n clients and an aggregator to collaboratively build a global model G. In a training iteration $t \in \{1,\ldots,T\}$, each client $i \in \{1,\ldots,n\}$ locally trains a local model W_i with p parameters (indicating both weights and biases) $w_i^1,\ldots,w_i^p$ based on the previous global model G_{t-1} using its local data D_i and sends it to the aggregator which aggregates the received models W_i into the global model G_t.

Several aggregation mechanisms have been proposed recently: 1) *Federated Averaging* (FedAvg) [38], 2) *Krum* [9], 3) *Adaptive Federated Averaging* [42], and 4) *Trimmed mean or median* [60]. Although we evaluate FLAME's effectiveness on several aggregation mechanisms in §7.1, we generally focus on FedAvg in this work as it is commonly applied in FL [21,28,39,44,47,50,54] and related work on backdoor attacks [7,22,51,57,59]. In FedAvg, the global model is updated by averaging the weighted models as follows: $G_t = \Sigma_{i=1}^{n} {}^{s_i \times W_i}/_{s}$, where $s_i = \|D_i\|, s = \Sigma_{i=1}^{n} s_i$. However, in practice, a malicious client might provide falsified information about its dataset size (i.e., a large number) to amplify the relative weight of its updates [57]. Previous works often employed equal weights ($s_i = 1/n$) for the contributions of all clients [7,51,59]. We adopt this approach in this paper, i.e., we set $G_t = \Sigma_{i=1}^{n} {}^{W_i}/_{n}$. Further, other state-of-the-art aggregation rules, e.g., *Krum* [9], *Adaptive Federated Averaging* [42], and *Trimmed mean or median* [60] also do not consider the sizes of local training datasets by design.

2.2 Backdoor Attacks on Federated Learning

In backdoor attacks, the adversary $\mathcal{A}$ manipulates the local models W_i of k compromised clients to obtain poisoned models W_i' that are then aggregated into the global model G_t and thus affect its properties. In particular, $\mathcal{A}$ wants the poisoned model G_t' to behave normally on all inputs except for specific attacker-chosen inputs $x \in I_{\mathcal{A}}$ (where $I_{\mathcal{A}}$ denotes the so-called *trigger set*) for which attacker-chosen (incorrect) predictions should be output. Figure 1 shows common techniques used in FL backdoor attacks, including 1) data poisoning, e.g., [45,51,59], where $\mathcal{A}$ manipulates training datasets of models, and 2) model poisoning, e.g., [7,57] where $\mathcal{A}$ manipulates the training process or the trained models themselves. Next, we will briefly discuss these attack techniques.

Data Poisoning. In this attack, $\mathcal{A}$ adds manipulated data $D^{\mathcal{A}}$ to the training datasets of compromised clients i by flipping data labels, e.g., by changing the labels of a street sign database so that pictures showing a 30 km/h speed limit are labeled as 80 km/h [51], or, by adding triggers into data samples (e.g., a specific pixel pattern added to images [59]) in combination with label flipping. We denote the fraction

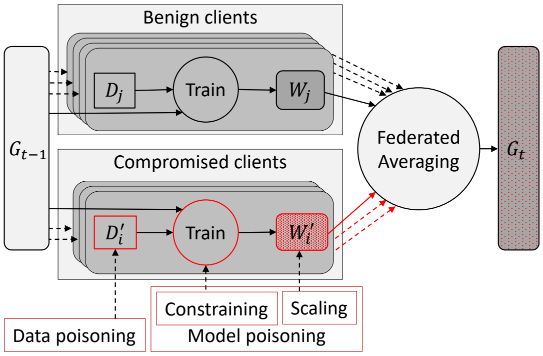


Figure 1: An overview of backdoor attacks.

of injected poisoned data $D_i^{\mathcal{A}}$ in the overall poisoned training dataset D'_i of client i as *Poisoned Data Rate (PDR)*, i.e., $PDR_i = |D_i^{\mathcal{A}}|/|D'_i|$.

Model Poisoning. This attack technique requires that $\mathcal{A}$ can fully control a number of clients. $\mathcal{A}$ poisons the training datasets of these clients and manipulates how they execute the training process by modifying parameters and scaling the resulting model update to maximize the attack impact while evading the aggregator's anomaly detector [7, 57]. This is done by (1) scaling up the weights of malicious model updates to maximize attack impact (e.g., *model-replacement* attack [7], or, *projected gradient descent (PGD) attack with model replacement* [57]), or, scaling down model updates to make them harder to detect (e.g., *train-and-scale* [7]) and (2) constraining the training process itself to minimize the deviation of malicious models from benign models to evade anomaly detection (e.g., *constrain-and-scale* attack [7]).

2.3 Adversary Goals and Capabilities

The goals of the adversary are two-fold:

Impact: The adversary $\mathcal{A}$ aims to manipulate the global model G so that the modified model G' provides incorrect predictions $f(G',x) = c' \neq f(G,x)$ for any inputs $x \in I_{\mathcal{A}}$, where $I_{\mathcal{A}}$ is the so-called *trigger set* consisting of specific attacker-chosen inputs and c' denotes the incorrect prediction chosen by the adversary.

Stealthiness: To make the poisoned model G' hard to detect by aggregator A, it should closely mimic the behavior of G on all other inputs not in $I_{\mathcal{A}}$, i.e.:

$$f(G',x) = \begin{cases} c' \neq f(G,x) & \forall x \in I_{\mathcal{A}} \\ \quad f(G,x) & \forall x \notin I_{\mathcal{A}} \end{cases} \tag{1}$$

Additionally, to make poisoned models as indistinguishable as possible from benign models, the distance (e.g., euclidean) between a poisoned model W' and a benign model W must be smaller than a threshold η denoting the distinction capability of the anomaly detector of aggregator A, i.e., $dist(W,W') < \eta$. The adversary can estimate this distance by comparing the local malicious model to the global model or to a local model trained on benign data.

Adversarial Capabilities. In this paper, we make no specific assumptions about the adversary's behavior. We assume that the adversary $\mathcal{A}$ has full control over $k < \frac{n}{2}$ clients and their training data, processes, and parameters [7, 59]. We denote the fraction of compromised clients as Poisoned Model Rate $PMR = \frac{k}{n}$. Furthermore, $\mathcal{A}$ has full knowledge of the aggregator's operations, including potentially applied backdoor defenses. However, $\mathcal{A}$ has no control over any processes executed at the aggregator nor over the honest clients.

2.4 Preliminaries

HDBSCAN [11] is a density-based clustering algorithm that uses the distance of data points in n-dimensional space to group data points that are located near each other together into a cluster. Hereby the number of clusters is determined dynamically. Data points that do not fit to any cluster are considered outliers. However, while HDBSCAN's predecessor DBSCAN [19] uses a predefined maximal distance to determine whether two points belong to the same cluster, HDBSCAN determines this maximal distance for each cluster independently, based on the density of points. Thus, in HDBSCAN, neither the maximal distance nor the total number of clusters need to be predefined.

Differential Privacy (DP). DP is a privacy technique that aims to ensure that the outputs do not reveal individual data records of participants. DP is formally defined as follows:

Definition 1 ((ε,δ)-differential privacy)**.** *A randomized algorithm $\mathcal{M}$ is (ε,δ)-differentially private if for any datasets D_1 and D_2 that differ on a single element, and any subset of outputs $\mathcal{S} \in Range(\mathcal{M})$, the following inequality holds:*

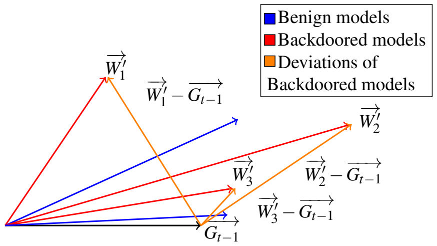
$$Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq e^{\varepsilon} \cdot Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta.$$

Here, ε denotes the privacy bound and δ denotes the probability of breaking this bound [18]. Smaller values of ε and δ indicate stronger privacy. A commonly used approach to enforce differential privacy is adding random Gaussian noise $N(0,\sigma^2)$ to the output of the algorithm [3, 18].

3 Problem Setting and Objectives

Backdoor Characterization. Following common practice in FL-related papers (e.g., [7, 12, 22]), we represent Neural Networks (NNs) using their weight vectors, in which the extraction of weights is done identically for all models by flattening/serializing the weight/bias matrices in a predetermined order. Figure 2 shows an abstract two-dimensional representation of the weight vectors of local models compared to the global model G_{t-1} of the preceding aggregation round. Each model W_i can be characterized with two factors: *direction (angle)* and *magnitude (length)* of its weight vector $(w^1, w^2, \ldots, w^p)$. The angle between two updates W_i and W_j can be measured, e.g., by using the cosine distance metric c_{ij} as defined in (2) while their magnitude difference is measured by the L$_2$-norm e_{ij} as defined in (3).

$$c_{ij} = 1 - \frac{W_i W_j}{\|W_i\| \, \|W_j\|} = 1 - \frac{\sum_{k=1}^{p} w_i^k w_j^k}{\sqrt{\sum_{k=1}^{p} (w_i^k)^2} \, \sqrt{\sum_{k=1}^{p} (w_j^k)^2}} \tag{2}$$



🟦 Benign models
🟥 Backdoored models
🟧 Deviations of Backdoored models
$\overrightarrow{W_1'}$
$\overrightarrow{W_1'} - \overrightarrow{G_{t-1}}$
$\overrightarrow{W_2'}$
$\overrightarrow{W_3'}$
$\overrightarrow{W_2'} - \overrightarrow{G_{t-1}}$
$\overrightarrow{W_3'} - \overrightarrow{G_{t-1}}$
$\overrightarrow{G_{t-1}}$

Figure 2: Weight vectors of benign and backdoored models.

$$e_{ij} = \left\|W_i - W_j\right\| = \sqrt{\sum_{k=1}^{p} \left(w_i^k - w_j^k\right)^2} \tag{3}$$

Benign and backdoored local models are shown in blue and red colors and are labeled with W_i or W_i', respectively. Note that the benign models are typically not identical due to the potentially partially non-iid nature of their training data.

The impact of the adversarial goal (injection of a backdoor) causes a deviation in the model parameters that manifests itself as a difference in the direction and/or magnitude of the backdoored model's weight vector in comparison to benign models, e.g., the deviations among local models and to the global model G_{t-1} of the previous aggregation round. Since the adversary has full control over the training process of compromised clients, he can fully control these distances, e.g., by changing the direction (in the case of W_1') or magnitude (in the case of W_2') of the backdoored models' weight vectors.

Figure 2 also shows three kinds of backdoored models resulting from different types of backdoor attacks. The first type W_1' has a similar weight vector, but a *large angular deviation* from the majority of local models and the global model. This is because such models are trained to obtain high accuracy on the backdoor task, which can be achieved by using a large poisoned data rate (*PDR*) or a large number of local training epochs (cf. Distributed Backdoor Attack (DBA) [59]). The second backdoor type W_2' has a small angular deviation but a *large magnitude* to amplify the impact of the attack. Such models can be crafted by the adversary by *scaling up* the model weights to boost its effect on the global model (cf. *Model-replacement* attack in [7]). The third backdoor type W_3' has a similar weight vector as benign models, the angular difference and the magnitude are not substantially different compared to benign models and, thus less distinguishable from benign models. Such *stealthy* backdoored models can be crafted by the adversary by carefully constraining the training process or scaling down the poisoned model's weights (cf. *Constrain-and-scale* attack [7] or FLIoT attack [45]).

Defense Objectives. A generic defense that can effectively mitigate backdoor attacks in the FL setting needs to fulfill the following objectives: (i) *Effectiveness*: To prevent the adversary from achieving its attack goals, the impact of backdoored model updates must be eliminated so that the aggregated global model does not demonstrate backdoor behavior. (ii) *Performance*: Benign performance of the global model must be preserved to maintain its utility. (iii) *Independence from data distributions and attack strategies*: The defense method must be applicable to generic adversary models, i.e., it must not require prior knowledge about the backdoor attack method, or make assumptions about specific data distributions of local clients, e.g., whether the data are iid or non-iid.

4 FLAME Overview and Design

We present the high-level idea of FLAME and the associated design challenges to fulfill the objectives identified in §3.

4.1 High-level Idea

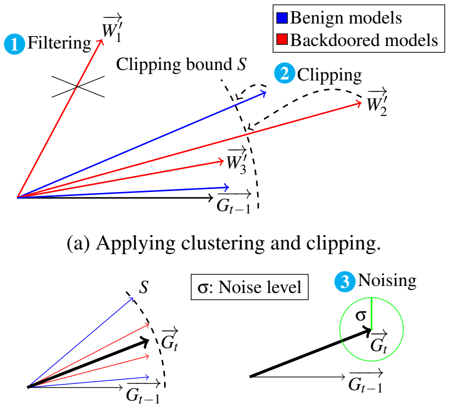
Motivation. Earlier works (e.g., Sun *et al.* [56]) use differential privacy-inspired noising of the aggregated model for eliminating backdoors. They determine the sufficient amount of noise to be used empirically. In the FL setting this is, however, challenging, as one cannot in general assume the aggregator to have access to training data, in particular to poisoned datasets. What is therefore needed is a generic method for determining how much noise is sufficient to remove backdoors effectively. On the other hand, the more noise is injected into the model, the more its benign performance will be impacted.

FLAME Overview. FLAME estimates the noise level required for backdoor removal in the FL setting without extensive empirical evaluation and having access to training data (this noise bound is formally proven in §5). In addition, to effectively limit the amount of required noise, FLAME uses a novel clustering-based approach to identify and remove adversarial model updates with high impact and applies a dynamic weight-clipping approach to limit the impact of models that the adversary has scaled up to boost their performance. As discussed in §3, one cannot guarantee that all backdoored models can be detected since the adversary can fully control both the angular and magnitude deviation to make the models arbitrarily hard to detect. Our clustering approach therefore aims to remove models with high attack impact (having larger angular deviation) rather than all malicious models. Fig. 3 illustrates the high-level idea of FLAME consisting of the above three components: filtering, clipping, and noising. We emphasize, however, that each of these components needs to be applied with great care, since, a naïve combination of noising with clustering and clipping leads to poor results as it easily fails to mitigate the backdoor and/or deteriorates the benign performance of the model, as we show in §C. We detail the design of each component and its use in the FLAME defense approach in §4.3.

4.2 Design Challenges

To realize the high-level idea presented above, we need to solve the following technical challenges.

C_1**- Filtering out backdoored models with large angular deviations in dynamic scenarios.** As discussed in §3, the weight vector of a well-trained backdoored model, W', has a *higher angular difference* in comparison to weight vectors of benign models W. FLAME deploys a clustering approach to



(a) Applying clustering and clipping.

(b) Applying noising

Figure 3: High-level idea of FLAME defense.

identify such poisoned models and *remove* them from FL aggregation (detailed in §4.3.1). The effect of clustering-based filtering is shown in Fig. 3a where model W_1' is removed from the aggregated model as it does not align with the directions of benign models. In contrast to existing clustering-based defenses, we need an approach that can also work in a *dynamic attack setting*, i.e., the number of injected backdoors is unknown and may vary between training rounds. To this end, we make a key observation: clustering approaches using a fixed number of clusters $n_{cluster}$ for identifying malicious models are inherently vulnerable to attacks with varying numbers of backdoors[2] $n_{backdoor}$. This is because the adversary can likely cause at least one backdoor model to be clustered together with benign models due to the pigeonhole principle by simultaneously injecting $n_{backdoor} \geq n_{cluster}$ backdoors. We seek to solve this challenge by employing a clustering solution that dynamically determines the clusters for model updates, thereby allowing it to adapt to dynamic attacks.

C_2**-Limiting the impact of scaled-up backdoors.** To limit the impact of backdoored models that the adversary artificially scales up to boost the attack (e.g., W_2' in Fig. 2), the weight vectors of models with high magnitudes can be clipped [56]. The effect of clipping is shown in Fig. 3a where the weight vectors of all models with a magnitude beyond the clipping bound S (in particular, backdoored model W_2') are clipped to S by scaling down the weight vectors. The resulting clipped weight vectors are shown on the left side of Fig. 3b. The challenge here is how to select a proper clipping bound without empirically evaluating its impact on the training datasets (which are not available in the FL setting). If the applied clipping bound is too large, an adversary can boost its model W' by scaling its weights up to the clipping bound, thereby maximizing the backdoor impact on the aggregated global model G. However, if the applied clipping bound is too small, a large fraction of benign model updates W will be clipped, thereby leading to performance deterioration of the aggregated global model G on the main task. We tackle this challenge in §4.3.2, where we show how to select a clipping bound that can not be influenced by the adversary and that effectively limits the impact of scaled-up backdoored models.

C_3**-Selecting suitable noise level for backdoor elimination.** As mentioned in §4.1, FLAME uses model noising that applies Gaussian noise with noise level σ to mitigate the adversarial impact of backdoored models (e.g., W_3' in Fig. 2). Similar to the clipping bound, however, also here the noise level σ must be carefully selected, as it has a direct impact on the effectiveness of the defense and the model's benign performance. If it is too low, the aggregated model might retain backdoor behavior after model noising, rendering the defense ineffective, while excessive noise will degrade the utility of the aggregated model. To address this challenge, we develop an approach for reliably estimating a sufficient but minimal bound for the applied noise in §5.

4.3 FLAME Design

As discussed in §4.1, our defense consists of three main components: filtering, clipping, and noising. Figure 4 shows these components and the workflow of FLAME during training round t. Algorithm 1 outlines the procedure of FLAME. In the rest of this section, we detail the design of these components to resolve the challenges in §4.2.

Algorithm 1 FLAME

```
Input: $n, G_0, T$   ▷ $n$ is the number of clients, $G_0$ is the initial global model, $T$ is the number of training iterations
Output: $G_T^*$ ▷ $G_T^*$ is the updated global model after $T$ iterations
for each training iteration $t$ in $[1,T]$ do
    for each client $i$ in $[1,n]$ do
        $W_i \leftarrow$ ClientUpdate($G_{t-1}^*$)   ▷ The aggregator sends $G_{t-1}^*$ to Client $i$ who trains $G_{t-1}^*$ using its data $D_i$ locally to achieve local modal $W_i$ and sends $W_i$ back to the aggregator.
    $(c_{11},\dots,c_{nn}) \leftarrow$ CosineDistance($W_1,\dots,W_n$)   ▷ $\forall i,j \in (1,\dots,n)$, $c_{ij}$ is the cosine distance between $W_i$ and $W_j$
    $(b_1,\dots,b_L) \leftarrow$ Clustering($c_{11},\dots,c_{nn}$)   ▷ $L$ is the number of admitted models, $b_l$ is the index of the $l^{th}$ model
    $(e_1,\dots,e_n) \leftarrow$ EuclideanDistances($G_{t-1}^*, (W_1,\dots,W_n)$)   ▷ $e_i$ is the Euclidean distance between $G_{t-1}^*$ and $W_i$
    $S_t \leftarrow$ Median($e_1,\dots,e_n$)   ▷ $S_t$ is the adaptive clipping bound at round $t$
    for each client $l$ in $[1,L]$ do
        $W_{b_l}^c \leftarrow G_{t-1} + (W_{b_l} - G_{t-1}) \cdot$ Min$(1,\gamma)$   ▷ Where $\gamma$ $(= S_t/e_{b_l})$ is the clipping parameter, $W_{b_l}^c$ is the admitted model after clipped by the adaptive clipping bound $S_t$
    $G_t \leftarrow \sum_{l=1}^{L} W_{b_l}^c / L$   ▷ Aggregating, $G_t$ is the plain global model before adding noise
    $\sigma \leftarrow \lambda \cdot S_t$ where $\lambda = \frac{1}{\varepsilon} \cdot \sqrt{2ln\frac{1.25}{\delta}}$   ▷ Adaptive noising level
    $G_t^* \leftarrow G_t + N(0,\sigma^2)$   ▷ Adaptive noising
```

4.3.1 Dynamic *Model Filtering*

The *Model Filtering* component of FLAME utilizes a *dynamic clustering* technique based on HDBSCAN [11] that

[2] We consider two backdoors to be independent if they use different triggers.

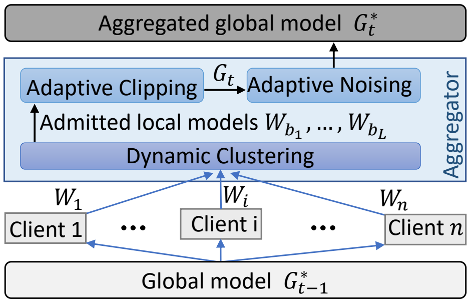
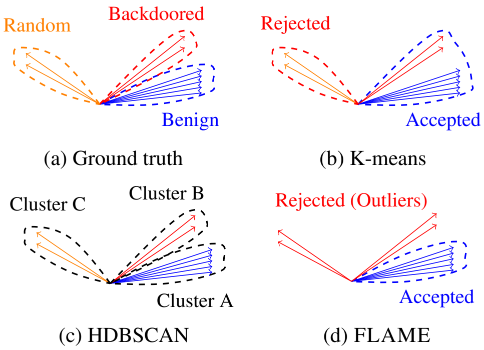


Figure 4: Illustration of FLAME's workflow in round t.

identifies poisoned models with high angular deviations from the majority of updates (e.g., W_1' in Fig. 3a). Existing clustering-based defenses [9, 51] identify potentially malicious model updates by clustering them into two groups where the smaller group is always considered malicious and thus removed. However, if no malicious models are present in the aggregation, this approach may lead to many models being incorrectly removed and thus a reduced accuracy of the aggregated model. These approaches also do not protect against attacks in which adversary $\mathcal{A}$ simultaneously injects multiple backdoors by using different groups of clients to inject different backdoors. If the number of clusters is fixed, there is the risk that poisoned and benign models end up in the same cluster, in particular, if models with different backdoors differ significantly. Consequently, existing model clustering methods do not adequately address challenge C_1 (§4.2). Fig. 5 shows the behavior of different clustering methods on a set of model updates' weight vectors. Fig. 5a shows the ground truth of an attack scenario where $\mathcal{A}$ uses two groups of clients: one group is used to inject a backdoor, whereas the other group provides random models with the goal of fooling clustering-based defenses. Fig. 5b shows how in this setting, K-means (as used in Auror [51]) fails to successfully separate benign and poisoned models as all poisoned models end up in the same cluster with the benign models.

To overcome the limitations of existing defenses, we design our clustering solution and ensure that: (i) it is able to handle dynamic attack scenarios where multiple backdoors are injected simultaneously, and (ii) it minimizes false positives of poisoned model identification. In contrast to existing approaches that try to place poisoned models into one cluster, our approach considers each poisoned model individually as an outlier, so that it can gracefully handle multiple simultaneous backdoors and thus address challenge C_1.

FLAME uses pairwise cosine distances to measure the angular differences between all model updates and applies the HDBSCAN clustering algorithm [11]. The advantage here is that cosine distances are not affected even if the adversary scales up model updates to boost their impact as this does not change the angle between the updates' weight vectors. Since the HDBSCAN algorithm clusters the models based on their density of the cosine distance distribution and *dynamically determines the required number of clusters*, we leverage it for our dynamic clustering approach. We describe HDBSCAN and how we apply it in detail in §E. In particular, HDBSCAN labels models as outliers if they do not fit into any cluster. This allows FLAME to effectively handle multiple poisoned models with different backdoors by labeling them as outliers. To realize this, we set the minimum cluster size to be at least 50% of the clients, i.e., $\frac{n}{2}+1$, so that the resulting cluster will contain the majority of updates (which we assume to be benign, cf. §2.3). All remaining (potentially poisoned) models are marked as *outliers*. This behavior is depicted in Fig. 5d where all the models from Clusters B and C from Fig. 5c are considered as outliers. Hence, to the best of our knowledge, our approach is the first FL backdoor defense that is able to gracefully handle also dynamic attacks in which the number of injected backdoors may vary. The clustering step is shown in lines 6-7 of Alg. 1 where L models are retained after clustering.



(a) Ground truth (b) K-means

(c) HDBSCAN (d) FLAME

Figure 5: Comparison of clustering quality for (a) ground truth, (b) using K-means with 2 clusters as in Auror [51], (c) straightforward applied HDBSCAN and (d) our approach as in FLAME.

4.3.2 Adaptive Clipping and Noising

As discussed in §4.2 (challenges C_2 and C_3), determining a proper clipping bound and noise level for model weight clipping and noising is not straightforward. We present our new approach for selecting an effective clipping bound and reliably estimating a sufficient noise level that can effectively eliminate backdoors while preserving the performance of the main task. Furthermore, our defense approach is resilient to adversaries that dynamically adapt their attacks.

Adaptive Clipping. Fig. 6 shows the variation of the average L$_2$-norms of model updates of benign clients in three different datasets (cf. §6) over subsequent training rounds. We can observe that the *L$_2$-norms of benign model updates become smaller in later training rounds*. To effectively remove backdoors while minimizing the impact on benign updates, the clipping bound S needs to be dynamically adapted to this decreasing trend of the L$_2$-norm. Recall that clipping is performed after clustering by scaling down model weights so that the L$_2$-norm of the scaled model becomes smaller or equal to the clipping threshold. We describe how FLAME determines a proper scaling factor for each model update W_i in t^{th} training round as follows: Given the index set $(b_1, \dots b_L)$ of the models admitted by the clustering method (line 7 of

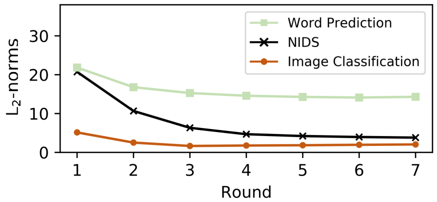


Figure 6: L_2-norms of model updates depending on the number of training rounds for different datasets.

Alg. 1), the aggregator first computes the clipping bound S_t as the median of the L_2-norms of all n model updates: $S_t = \textsc{Median}(e_1, \ldots, e_n)$. It should be noted that for determining the clipping bound, the rejected models are also considered to ensure that even if benign models were filtered, the computed median S_t is still determined based on benign values. However, after determining the clipping bound, only the admitted models $W_1, \ldots, W_L$ are considered for later processing. The scaling factor for the l^{th} admitted model is computed as $\gamma = \frac{S_t}{e_{b_l}}$ where e_{b_l} is the L_2-norm of the model update W_{b_l}. Clipping scales down model updates as follows: $W^c_{b_l} = G_{t-1} + (W_{b_l} - G_{t-1}) \cdot \textsc{Min}(1, \gamma)$ (detailed in line 8-11 of Alg. 1) where the multiplication is computed coordinate-wise. It is worth noting that weighting contributions (i.e., adjusting scaling factor) based on client data sizes is insecure. As we point out in §2.1, the reported dataset sizes by clients cannot be trusted, i.e., the adversary can lie about their dataset sizes to maximize attack impact [57]. Hence, we follow common practice in literature and weight the contributions of all clients equally regardless of their dataset size [7, 9, 12, 59]. By using the median as the clipping bound S_t, we ensure that S_t is always in the range of the L_2-norms between benign models and the global model since we assume that more than 50% of clients are benign (cf. §2.3). We evaluate the effectiveness of the clipping approach in §B.2.

Adaptive Noising. It has been shown that by adding noise to a model's weights, the impact of outlier samples can be effectively mitigated [17]. Noise can also be added to poisoned samples (special cases of outliers) used in backdoor injection. The more noise is added to the model during the training process, the less responsive the model will be to the poisoned samples. Thus, increasing model robustness against backdoors. Eliminating backdoors utilizing noise addition is conceptually the same in a centralized or federated setting (e.g., [7, 17]): In both cases, noise is added to the model weights to smooth out the effect of poisoned data (cf. Eq. 5). The challenge is to determine as small a noise level as possible to eliminate backdoors and at the same time not deteriorate the benign performance of the model. As we discuss in detail in §5.1, the amount of noise is determined by estimating the sensitivity based on the differences (distances) among local models, which can be done without access to training data. We then add Gaussian noise to the global model G_t to yield a noised global model G^*_t as follows: $G^*_t = G_t + N(0, \sigma^2)$, see Lines 13-14 of Alg. 1 for more details. This ensures that backdoor contributions are effectively eliminated from the aggregated model. In particular, we show in §5.1 how the noise-based backdoor elimination technique can be transferred from a centralized to a federated setting by analysing the relationship between aggregated Gaussian noise applied to the global model and individual noising of each local model.

5 Security Analysis

5.1 Noise Boundary Proof of FLAME

In this section, we provide a proof to corroborate that FLAME can neutralize backdoors in the FL setting by applying strategical noising with bound analysis on the noise level. We first formulate the noise boundary guarantee of FLAME in Theorem 1. Subsequently, we explain related parameters and prove how the noise level bound for σ can be estimated. This is done by generalizing theoretical results from previous works [17, 18] to the FL setting. Then, we show how the filtering and clipping component of FLAME helps to effectively reduce the noise level bound in Theorem 2. We provide a formal proof for linear models and extend the proof to DNNs using empirical evaluation. This is because providing formal proof for DP-based backdoor security for DNN models is still an open research problem even for centralized settings.

Theorem 1. *A (ε, δ)-differentially private model with parameters G and clipping bound S_t is backdoor-free if random Gaussian noise is added to the model parameters yielding a noised version G^* of the model: $G^* \leftarrow G + N(0, \sigma^2_G)$ where the noise scale σ_G is determined by the clipping bound S_t and a noise level factor λ: $\sigma_G \leftarrow \lambda \cdot S_t$ and $\lambda = \frac{1}{\varepsilon} \cdot \sqrt{2ln\frac{1.25}{\delta}}$.*

We explore the key observation that an ML model with a sufficient level of differential privacy is *backdoor-free*. With this new definition of backdoor-free models in the DP domain, the main challenge to defeat backdoors in the FL setting is to decide a proper noise scale for the global model without knowledge of the training datasets. Furthermore, we need to minimize the amount of noise added to the global model to preserve its performance on the main task. None of the prior DP-based FL backdoor defense techniques provide a solution to the noise determination problem [56]. For the first time, FLAME presents an approach to *estimate* the proper noise scale that ensures the global model is backdoor-free. The noise boundary proof in Theorem 1 consists of two steps:
Step 1 (S1). By introducing the data hiding property of DP (Def. 1) and its *implication as the theoretical guarantee for backdoor-free* models. We also discuss function sensitivity (Def. 2) which is an important factor for selection of the DP parameters (ε, δ).
Step 2 (S2). We show how FLAME *generalizes backdoor elimination from centralized setting to federated setting* with theoretical analysis of the noise boundary (Eq. 5 and 6). FLAME *is the first FL defense against backdoors that provides noise level proof with bounded backdoor effectiveness.*

(S1) DP foundations and re-interpretation as Backdoor-free. As discussed in §2.4, by definition, DP makes the difference between data points indistinguishable. FLAME leverages this property of DP for backdoor elimination. In particular, we can consider D_1 and D_2 in Def. 1 as the benign and backdoored dataset. The inequality of DP suggests that algorithm $\mathcal{M}$ has a high probability of producing the same outputs on the benign and the poisoned dataset, meaning that the backdoor is eliminated. The noise level σ is determined based on the DP parameters (ε, δ) and the *sensitivity* of the function f defined below:

Definition 2 (Sensitivity)**.** *Given the function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ where $\mathcal{D}$ is the data domain and d is the dimension of the function output, the sensitivity of the function f is defined as:*

$$\Delta = \max_{D_1,\, D_2\, \in\, \mathcal{D}} ||f(D_1) - f(D_2)||_2, \tag{4}$$

where D_1 and D_2 differs on a single element $||D_1 - D_2||_1 = 1$.

As shown in Lemma 1 [18], this definition can be extended to datasets differing by more than one element, i.e., can be generalized to the DP in the multiple-point-difference setting.

(S2) Generalizing backdoor resilience from centralized to federated setting (FLAME). In the centralized setting, the defender has access to the model to be protected, the benign dataset, and the outlier (backdoored) samples. As such, he can estimate the sensitivity Δ for (ε, δ)-DP. When applying Gaussian noise with the noise scale $\sigma = \frac{\Delta}{\varepsilon}\sqrt{2ln\frac{1.25}{\delta}}$, the defender can enforce a lower bound on the prediction loss of the model on the backdoored samples for backdoor elimination [28]. However, this robustness rationale *cannot* be directly transferred from the centralized setting to the FL setting since the defender in the federated scenario (i.e., aggregator) only has access to received model updates, but not the datasets to estimate the sensitivity Δ for the global model.

FLAME extends DP-based noising for backdoor elimination to the federated setting based on the following observation: if one can ensure that all aggregated models are benign (i.e., backdoor-free), then it is obvious that the aggregated global model will also be backdoor-free. This intuition can be formally proven if the FL aggregation rule is Byzantine-tolerant. To ensure that any backdoor potentially present in the model is eliminated and the aggregated model is benign, a sufficient DP noise level is added to individual local models. However, since the local models are independent, adding noise to each local model is mathematically equivalent to the case where aggregated noise is added to the global model. This is conceptually equivalent to the conventional centralized setting, for which it has been formally shown that DP noise can eliminate backdoors [17]. In the following, we therefore show that adding DP noise to local models is equivalent to adding 'aggregated' DP noise to the global model.

We write the standard deviation of noise for the local models in the form $\sigma_i \leftarrow \frac{\alpha_i \cdot e_i}{\varepsilon} \cdot \sqrt{2ln\frac{1.25}{\delta}}$ where $\alpha_i = \frac{\Delta_i}{e_i}$, Δ_i and e_i is the sensitivity and the L_2 norm of the model W_i, respectively. Mathematically, the FL system with FLAME has:

$$\begin{aligned} G^* &= \frac{1}{n}\Sigma_{i=1}^n W_i^* = \frac{1}{n}[\,\Sigma_{i=1}^n\ W_i + N(0,\sigma_i^2)] \\ &= \frac{1}{n}\ \Sigma_{i=1}^2 W_i + \frac{1}{n}\Sigma_{i=1}^n N(0,\sigma_i^2) \\ &= \frac{1}{n}\ \Sigma_{i=1}^2 W_i + N(0, \frac{1}{n}\Sigma_{i=1}^n \sigma_i^2) \\ &= G + N(0,\sigma_G^2) \end{aligned} \tag{5}$$

in which W_i^* are local models and G^* the global model after adding noise $N(0,\sigma_i^2)$. Equation 5 represents the fact that adding DP noise to each local model (i.e., $W_i + N(0,\sigma_i^2)$) is equivalent to adding an 'aggregated' DP noise on the global model (i.e., $G + N(0,\sigma_G^2)$). More specifically, this equivalent Gaussian noise on the global model is the sum of Gaussian noise applied on each local model with a scaling factor $N_G = \frac{1}{n}\Sigma_{i=1}^n N_i$. Here, N_G and N_i are random variables with distribution $N(0,\sigma_G^2)$ and $N(0,\sigma_i^2)$, respectively. As such, we can compute the equivalent noise scale for the global model:

$$\begin{aligned} \sigma_G^2 &= \frac{1}{n^2}\Sigma_{i=1}^n \sigma_i^2 = (\frac{1}{\varepsilon}\sqrt{2ln\frac{1.25}{\delta}})^2 \cdot \frac{1}{n^2}\Sigma_{i=1}^n \Delta_i^2 \\ &= (\frac{1}{\varepsilon}\sqrt{2ln\frac{1.25}{\delta}})^2 \cdot \frac{1}{n^2}\Sigma_{i=1}^n \alpha_i^2 e_i^2. \end{aligned} \tag{6}$$

Equation 6 describes the relation between the DP noise added on FLAME's global model and the DP noise added on each local model. This noise scale relation in Eq. 6 together with the transformation in Eq. 5 enable FLAME to provide guaranteed security for the global model against backdoors, thereby addressing Challenge C_3 .

In Alg. 1, we use the median of Euclidean distances e_i as the upper bound S_t to clip the admitted local models (line 9-11). We hypothesize that the sensitivity of a model W_i is positively correlated with its weight magnitude $|W_i|$ (see Theorem 2 for details). In the case of linear models, the sensitivity Δ has a *linear* relation with the model weight $|\overrightarrow{w}|$ (see Eq. 8). Therefore, we use the following approximation:

$$\frac{1}{n^2}\Sigma_{i=1}^n \alpha_i^2 e_i^2 = \frac{1}{n^2}\Sigma_{i=1}^n \Delta_i^2 \approx S_t^2,$$

where S_t is the weight clipping bound. Having substituted the above approximation into Eq. 6, we can compute the noise scale of DP that FLAME deploys on the global model N_G:

$$\sigma_G \approx \frac{S_t}{\varepsilon}\sqrt{2ln\frac{1.25}{\delta}} \tag{7}$$

This concludes the proof of Theorem 1. □

FLAME's adaptive noising step applies the Gaussian noise with the noise scale computed in Eq. 7 on the global model for backdoor elimination as shown in Alg. 1, line 13-14. Note that FLAME's noising scheme is *adaptive* since the clipping bound S_t is obtained dynamically in each t^{th} epoch.

Next, we present Theorem 2 and justify how FLAME design reduces the derived noise level with *step 3 (S3)* below.

(S3) Clustering and clipping components in FLAME help to reduce the DP noise boundary. Recall that FLAME protects the FL system against backdoor attacks using three steps: clustering, clipping, and adding DP noise. The overall workflow of FLAME is shown in Fig. 4. If multiple backdoors exist in the FL system, the first two steps (clustering and clipping) can remove a subset of backdoors as shown in Fig. 3a. Note that the remaining backdoors are *'closer'* to the benign model updates in terms of both magnitude and direction. This gives us the *intuition* that removing the remaining backdoors by adding DP noise becomes easier (i.e., the noise scale σ_G is smaller) after the first two steps of FLAME.

We can see from Theorem 1 that the Gaussian noise scale σ required for backdoor resilience increases with the sensitivity of each local model Δ_i. We describe two characteristics of the model parameter W, i.e., direction and magnitude in §4. We discuss how these two factors impact the sensitivity of the model defined in Eq. 4 below.

Theorem 2. *Backdoor models with large angular deviation from benign ones, or with large parameter magnitudes have high sensitivity values* Δ.

Proving DP-based backdoor security for DNN models is still an open problem, even in the centralized setting. We, therefore, adopt a common approach in literature (e.g., [17]) by providing theoretical proof for linear models and validating it for DNNs empirically.

Proof: for a linear model f where the function output is determined by the inner product of model weight vector $\overrightarrow{w}$ and the data vector $\overrightarrow{x}$, we have

$$f(w;\, x) = \overrightarrow{w} \cdot \overrightarrow{x} = |w| \cdot |x| \cdot cos\theta, \tag{8}$$

where $\theta =< \overrightarrow{w}, \overrightarrow{x} >$ is the angle between two vectors. In this case, it is straightforward to see that if the backdoor attack changes the parameter magnitude $|w|$ or the direction θ of the model f, the resulting poisoned model f' has a large sensitivity value based on the definition in Eq. 4. □

This analysis suggests that backdoor models with large angular deviations or with large weight magnitudes have a high sensitivity value Δ. Recall that FLAME deploys dynamic clustering (§4.3.1) to remove poisoned models with large cosine distances, and employs adaptive clipping (§4.3.2) to remove poisoned models with large magnitudes. Therefore, the sensitivity of the remaining backdoor models is lower compared to the one before applying these two steps. As a result, FLAME *can use a small Gaussian noise to eliminate the remaining backdoors after applying clustering and clipping, which is beneficial for preserving the main task accuracy.*

We empirically show how the noise scale for backdoor elimination changes after applying each step of FLAME. Particularly, we measure the *smallest* Gaussian noise scale σ required to defeat *all backdoors* (i.e., $BA = 0\%$) in three settings: i) No defense components applied (which is equivalent to the previous DP-based defense [7, 18]); ii) After applying dynamic clustering; iii) After applying both dynamic clustering and adaptive clipping (which is the setting of FLAME).

Table 1: Effect of clustering and clipping in FLAME on minimal Gaussian noise level σ for backdoor elimination in the NIDS scenario, in terms of Backdoor Accuracy (*BA*) and Main Task Accuracy (*MA*).

σ	Only Noising		After Clustering		After Clustering & Clipping	
	BA	MA	BA	MA	BA	MA
0.01	100.0%	100.0%	0.0%	80.5%	0.0%	100.0%
0.08	3.5%	66.7%	0.0%	66.7%	0.0%	100.0%
0.10	0.0%	54.2%	0.0%	66.1%	0.0%	87.6%

We conduct this comparison experiment on the IoT-Traffic dataset (cf. §6). For each communication round, 100 clients are selected where $k = 40$ are adversaries. We remove the backdoor by adding Gaussian noise $N(0, \sigma^2)$ to the aggregated model. Table 1 summarizes the evaluation results in the above three settings. We can observe from the comparison results that the noise scale required to eliminate backdoors decreases after individual deployment of clustering and clipping. This corroborates the correctness of Theorem 2.

5.2 Attack and Data Distribution Assumption

In FLAME, we do not make specific assumptions about the attack and data distribution compared to the existing clustering-based defenses. Let $X = (X_1, \ldots, X_b)$ be a set of distributions of benign models $(W_1, \ldots, W_{n-k})$ where $b \leq n - k$. The deviation in X is caused by the diversity of the data. Let $X' = (X'_1, \ldots, X'_a)$ be a set of distributions of poisoned models $(W'_1, \ldots, W'_k)$ where $a \leq k$. The deviation in X' is caused by the diversity of the benign data and backdoors (e.g., poisoned data or model crafting). Existing works assume that $X'_i \approx X'_j \quad (\forall i, j : 1 \leq i, j \leq a)$ (see e.g., [22] or $X' \neq X$ [9, 51]). However, this assumption does not hold in many situations because (i) there can be one or multiple attackers injecting multiple backdoors [7], or (ii) the adversary can inject one or several random (honeypot) models having a distribution X'_r that is significantly different from $X \cup (X' \setminus X'_r)$, and (iii) the adversary can control how much the backdoored models deviate from benign ones as discussed in §3. Therefore, approaches that purely divide models into two groups, e.g., K-means [51] will incorrectly classify models having distribution X'_r into the malicious group and all remaining models (having distributions drawn from $(X \cup (X' \setminus X'_r))$ into the benign group. As a result, all backdoored models having distributions drawn from $(X' \setminus X'_r)$ are classified as benign, as demonstrated in Fig. 5b. In contrast, FLAME does not rely on such specific assumptions (the adversary can arbitrarily choose X'). If the distribution X'_i of a poisoned model is similar to benign distributions in X, FLAME will falsely classify X'_i as being. But if the distribution X'_j of a poisoned model is different from the distributions in X, FLAME will identify X'_j as an outlier and classify the associated model as malicious. To identify deviating and thus potentially malicious models, FLAME leverages the HDBSCAN algorithm to identify regions of high density in the model space. Any models that are not located in the dense regions will be categorized as outliers, as shown in Fig. 5d. As discussed in §3, FLAME aims to remove models with distributions X'_j that have a higher attack impact compared to models with distribution X'_i. It is worth noting, however, that the impact of such remaining backdoored models will be eliminated by the noising component as shown in §5.1

Striking a balance between accuracy and security: Clustering and DP-based approaches affect model accuracy as discussed in §4.2 (Challenges C_2 and C_3). In particular, an approach that aims to maximize the number of filtered malicious models may lead to many false positives, i.e., many benign models being filtered out. Moreover, applying a very low clipping bound or a very high level of injected noise will degrade model accuracy. To address these problems, FLAME is configured so that the clustering component removes only models with high attack impact rather than all malicious models, i.e., it aims to remove the first backdoor type W'_1 as shown in Fig. 3. In addition, FLAME carefully estimates the clipping bound and noise level to ensure backdoor elimination while preserving model performance. As discussed in §4.3.2, the L_2-norms of model updates depend on the number of training rounds, dataset types, and type of backdoors. Consequently, the clipping threshold and noise level should be adapted to L_2-norms. We therefore apply the median of the L_2-norms of model updates as the clipping bound S_t (cf. Lines 9-11 of Alg. 1). This ensures that S_t is always computed between a benign local model and the global model since we assume that more than 50% of clients are benign (cf. §2.3). Further, estimating noise level based on S_t (cf. Lines 13-14 of Alg. 1) also provides a noise boundary that ensures that the global model is resilient against backdoors as discussed in §5.1. Moreover, our comparison of potential values for S_t presented in §B.2 and §B.3 shows that the chosen clipping bound and noise level provide the best balance between accuracy and security, i.e., FLAME eliminates backdoor while retaining the global model's performance on the main task.

6 Experimental Setup

We conduct all the experiments using the PyTorch deep learning framework [2] and use the source code provided by Bagdasaryan *et al.* [7], Xie *et al.* [59] and Wang *et al.* [57] to implement the attacks. We reimplemented existing defenses to compare them with FLAME.

Datasets and Learning Configurations. Following recent research on poisoning attacks on FL, we evaluate FLAME in three typical application scenarios: word prediction [35, 38–40], image classification [13, 49, 50], and an IoT intrusion detection [44, 47, 48, 54] as summarized in Tab. 2. Verification of the effectiveness of FLAME against state-of-the-art attacks in comparison to existing defenses (cf. Tab. 3 and Tab. 4) are conducted on these three datasets in the mentioned application scenarios. Experiments for evaluating specific performance aspects of FLAME are performed on the IoT dataset as it represents a very diverse and real-world setting with clear security implications.

Table 2: Datasets used in our evaluations.

Application	Datasets	#Records	Model	#params
WP	Reddit	20.6M	LSTM	$\sim$20M
NIDS	IoT-Traffic	65.6M	GRU	$\sim$507k
IC	CIFAR-10	60k	ResNet-18 Light	$\sim$2.7M
	MNIST	70k	CNN	$\sim$431k
	Tiny-ImageNet	120k	ResNet-18	$\sim$11M

Evaluation Metrics. We consider a set of metrics for evaluating the effectiveness of backdoor attack and defense techniques as follows: *BA - Backdoor Accuracy* indicates the accuracy of the model in the backdoor task, i.e., it is the fraction of the trigger set for which the model provides the wrong outputs as chosen by the adversary. The adversary aims to maximize *BA*, while an effective defense prevents the adversary from increasing it. *MA - Main Task Accuracy* indicates the accuracy of a model in its main (benign) task. It denotes the fraction of benign inputs for which the system provides correct predictions. The adversary aims at minimizing the effect on *MA* to reduce the chance of being detected. The defense system should not negatively impact *MA*. *TPR - True Positive Rate* indicates how well the defense identifies poisoned models, i.e., the ratio of the number of models correctly classified as poisoned (True Positives - TP) to the total number of models being classified as poisoned: $TPR = \frac{TP}{TP+FP}$, where FP is False Positives indicating the number of benign clients that are wrongly classified as malicious. *TNR - True Negative Rate* indicates the ratio of the number of models correctly classified as benign (True Negatives - TN) to the total number of benign models: $TNR = \frac{TN}{TN+FN}$, where FN is False Negatives indicating the number of malicious clients that are wrongly classified as benign.

7 Experimental Results

In this section, we evaluate FLAME against backdoor attacks in the literature (§7.1) and demonstrate that our defense mechanism is resilient to adaptive attacks (§7.2). In addition, we show the effectiveness of each of FLAME's components in §B and FLAME overhead in §D. Finally, we evaluate the impact of the number of clients (§7.3) as well as the degree of non-IID data (§7.4).

7.1 Preventing Backdoor Attacks

Effectiveness of FLAME. We evaluate FLAME against the state-of-the-art backdoor attacks called *constrain-and-scale* [7], DBA [59], PGD and Edge-Case [57] and an untargeted poisoning attack [20] (cf. §F) using the same attack settings as in the original works with multiple datasets. The results are shown in Tab. 3. FLAME completely mitigates the *constrain-and-scale* attack ($BA = 0\%$) for all datasets. Moreover, our defense does not affect the Main Task Accuracy (*MA*) of the system as MA reduces by less than 0.4% in all experiments. The DBA attack as well as the Edge-Case attack [57] are also successfully mitigated ($BA = 3.2\%/4.0\%$). Further, FLAME is also effective against PGD attacks ($BA =$

Table 3: Effectiveness of FLAME against state-of-the-art attacks for the respective dataset, in terms of Backdoor Accuracy (*BA*) and Main Task Accuracy (*MA*). All metric values are reported as percentages.

Attack	Dataset	No Defense		FLAME	
		BA	*MA*	*BA*	*MA*
Constrain-and-scale [7]	Reddit	100	22.6	0	22.3
	CIFAR-10	81.9	89.8	0	91.9
	IoT-Traffic	100.0	100.0	0	99.8
DBA [59]	CIFAR-10	93.8	57.4	3.2	76.2
Edge-Case [57]	CIFAR-10	42.8	84.3	4.0	79.3
PGD [57]	CIFAR-10	56.1	68.8	0.5	65.1
Untargeted Poisoning [20]	CIFAR-10	-	46.72	-	91.31

Table 4: Effectiveness of FLAME in comparison to state-of-the-art defenses for the *constrain-and-scale* attack on three datasets, in terms of Backdoor Accuracy (*BA*) and Main Task Accuracy (*MA*). All values are percentages.

Defenses	Reddit		CIFAR-10		IoT-Traffic	
	BA	*MA*	*BA*	*MA*	*BA*	*MA*
Benign Setting	-	22.7	-	92.2	-	100.0
No defense	100.0	22.6	81.9	89.8	100.0	100.0
Krum [9]	100.0	9.6	100.0	56.7	100.0	84.0
FoolsGold [22]	**0.0**	**22.5**	100.0	52.3	100.0	99.2
Auror [51]	100.0	**22.5**	100.0	26.1	100.0	96.6
AFA [42]	100.0	22.4	**0.0**	91.7	100.0	87.4
DP [18]	14.0	18.9	**0.0**	78.9	14.8	82.3
Median [60]	**0.0**	22.0	**0.0**	50.1	0.0	87.7
FLAME	**0.0**	22.3	**0.0**	**91.9**	**0.0**	**99.8**

0.5 %). It should be noted that suggesting words is a quite challenging task, causing the MA even without attack to be only 22.7%, aligned with previous work [7].

We extend our evaluation to various backdoors on three datasets. For NIDS, we evaluate 13 different backdoors (Mirai malware attacks) and 24 device types (78 IoT devices). The results show that FLAME is able to mitigate all backdoor attacks completely while achieving a high MA=99.8%. We evaluate 5 different word backdoors for WP, and 90 different image backdoors for IC, which change the output of a whole class to another class. In all cases, FLAME successfully mitigates the attack while still preserving the *MA*.

Comparison to existing defenses. We compare FLAME to existing defenses: Krum [9], FoolsGold [22], Auror [51], Adaptive Federated Averaging (AFA) [42], Median [60] and a generalized differential privacy (DP) approach [7, 40]. Tab. 4 shows that FLAME is effective for all 3 datasets, while previous works either fail to mitigate backdoors or reduce the main task accuracy. Krum, FoolsGold, Auror, and AFA do not effectively remove poisoned models and *BA* often remains at 100%. Also, some defenses make the attack even more successful than without defense. Since they remove many benign updates (cf. §B) but fail to remove a sufficient number of poisoned updates, these defenses increase the *PMR* and, therefore, also the impact of the attack. Some defenses, e.g., Krum [9], Auror [51] or AFA [42] are not able to handle non-iid data scenarios like Reddit. In contrast, FoolsGold is only effective on the Reddit dataset ($TPR = 100\%$) because it works well on highly non-independent and identically distributed (non-IID) data (cf. §9). Similarly, AFA only mitigates backdoors on the CIFAR-10 dataset since the data are highly IID (each client is assigned a random set of images) such that the benign models share similar distances to the global model (cf. §9). Additionally, the model's *MA* is negatively impacted. The DP-based defense is effective, but it significantly reduces *MA*. For example, it performs best on the CIFAR-10 dataset with $BA = 0$, but *MA* decreases to 78.9% while FLAME increases *MA* to 91.9% which is close to the benign setting (no attacks), where $MA = 92.2\%$.

Effectiveness of FLAME's Components. Further, we have also conducted an extensive evaluation of the effectiveness of each of FLAME's components. Due to space limitations, we would like to refer to §B for the details.

7.2 Resilience to Adaptive Attacks

Given sufficient knowledge about FLAME, an adversary may seek to use adaptive attacks to bypass the defense components. In this section, we analyze such attack scenarios and strategies including *changing the injection strategy*, *model alignment*, and *model obfuscation*.

Changing the Injection Strategy. The adversary $\mathcal{A}$ may attempt to inject several backdoors simultaneously to execute different attacks on the system in parallel or to circumvent the clustering defense (cf. §2.2). FLAME is also effective against such attacks (cf. Fig. 5). To further investigate the resilience of FLAME against such attacks, we conduct two experiments: 1) assigning different backdoors to malicious clients and 2) letting each malicious client inject several backdoors. To ensure that each backdoor is injected by a sufficient number of clients, we increased the PMR for this experiment. We conducted these experiments with $n = 100$ clients of which $k = 40$ are malicious on the IoT-Traffic dataset with each type of Mirai attack representing a backdoor. First, we evaluate FLAME for $0, 1, 2, 4$, and 8 backdoors, meaning that the number of malicious clients for each backdoor is $0, 40, 20, 10$, and 5. Our experimental results show that our approach is effective in mitigating the attacks as $BA = 0\% \pm 0.0\%$ in all cases, with $TPR = 95.2\% \pm 0.0\%$, and $TNR = 100.0\% \pm 0.0\%$. For the second experiment, 4 backdoors are injected by each of the 40 malicious clients. Also, in this case, the results show that FLAME can completely mitigate the backdoors.

Model Alignment. Using the same attack parameter values, i.e., *PDR* (cf. §2.2), for all malicious clients can result in high distances between benign and poisoned models. Those high distances can be illustrated as a gap between poisoned and benign models, s.t. the clustering can separate them. Therefore, a sophisticated adversary can generate models that bridge the gap between them such that they are merged to the same cluster in our clustering. We evaluate this attack on the IoT-Traffic dataset for $k = 80$ malicious clients and $n = 200$ clients in total. To remove the gap, each malicious client is assigned a random amount of malicious data, i.e., a random *PDR* ranging from 5% to 20%. As Tab. 5 shows, when we apply model

Table 5: Resilience to model alignment attacks in terms of Backdoor Accuracy (*BA*), Main Task Accuracy (*MA*), True Positive Rate (*TPR*), True Negative Rate (*TNR*) in percent.

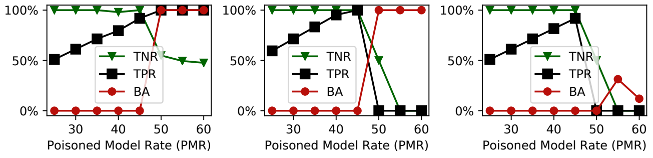
	BA	*MA*	*TPR*	*TNR*
Model Filtering	100.0	91.98	0.0	33.04
FLAME	**0.0**	**100.0**	**5.68**	**33.33**

filtering only, our clustering component cannot identify the malicious clients well ($TPR = 0\%$), resulting in $BA = 100\%$. However, when we apply FLAME, although *TPR* remains low (5.68%) FLAME still mitigates the attack successfully (*BA* reduces from 100% to 0%). This can be explained by the fact that when the adversary $\mathcal{A}$ tunes malicious updates to be close to the benign ones, the attack's impact is reduced and consequently averaged out by our noising component.

Model Obfuscation. $\mathcal{A}$ can add noise to the poisoned models to make them difficult to detect. However, our evaluation of such an attack on the IoT-Traffic dataset shows that this strategy is not effective. We evaluate different noise levels to determine a suitable standard deviation for the noise. Thereby, we observe that a noise level of 0.034 causes the models' cosine distances in clustering to change without significantly impacting *BA*. However, FLAME can still efficiently defend this attack: *BA* remains at 0% and *MA* at 100%.

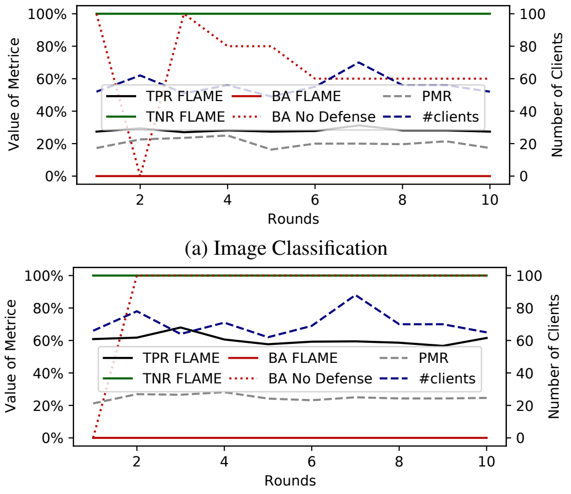
7.3 Effect of Number of Clients

Impact of Number of Malicious Clients. We assume that the number of benign clients is more than half of all clients (cf. §2.2) and our clustering is only expected to be successful when $PMR = \frac{k}{n} < 50\%$ (cf. §4.3.1). We evaluate FLAME for different *PMR* values. Figure 7 shows how *BA*, *TPR*, and *TNR* change in the IC, NIDS, and WP applications for *PMR* values from 25% to 60%. It shows that FLAME is only effective if $PMR < 50\%$ so that only benign clients are admitted to the model aggregation ($TNR = 100\%$) and thus $BA = 0\%$. However, if $PMR > 50\%$, FLAME fails to mitigate the attack because the majority of poisoned models will be included resulting in low *TNR*. Interestingly, FLAME accepted all models for $PMR = 50\%$ ($TPR = 0\%$ and $TNR = 100\%$). For the IC application, since the IC data are non-IID, poisoned models are not similar. Therefore, some poisoned models were excluded from the cluster resulting in a high *TPR* even for *PMRs* higher than 50%. However, the majority of poisoned models were selected resulting in the drop in the *TNR*.



(a) IC (b) NIDS (c) WP

Figure 7: Impact of the poisoned model rate $PMR = \frac{k}{n}$ on the evaluation metrics. *PMR* is the fraction of malicious clients *k* per total clients *n*.



(a) Image Classification

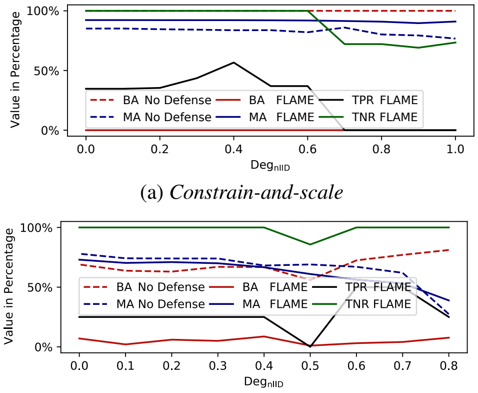
(b) Network Intrusion Detection

Figure 8: Impact of the number of clients on FLAME

Varying number of clients in different training rounds. In general, FLAME is a round-independent defense, i.e., it does not use information from previous rounds such as which clients were excluded in which rounds. Therefore, FLAME will not be affected if the number of clients or number of malicious clients varies as long as the majority of clients remain benign. To demonstrate this, we simulate realistic scenarios in which clients can join and drop out dynamically. We conducted an experiment where during each round, the total number of available clients is randomly selected. As the result, the number of malicious clients will also be random. In this experiment, we used a population of 100 clients in total, out of which 25 are malicious. In each round, a random number (from 60 to 90) of clients are selected, so that the fraction of malicious clients (*PMR*) varies in each round. Figure 8 shows the experimental results. One can see that the proportion of malicious clients (*PMR*) does not affect the effectiveness of FLAME, i.e., the backdoor is completely removed ($BA = 0\%$) in every round. Since all poisoned models are detected, their negative effect on the aggregated model is removed. Therefore, the MA with FLAME is better than the one without defense, and is almost always 100 % aligned with the results in Tab. 4.

7.4 Impact of the Degree of non-IID Data

Since clustering is based on measuring differences between benign and malicious updates, the distribution of data among clients might affect our defense. We conduct two experiments for both *Constrain-and-scale* and Edge-Case PGD on the CIFAR-10 dataset. For Reddit and IoT datasets, changing the degree of non-IID data is not meaningful since the data have a natural distribution as every client obtains data from different Reddit users or traffic chunks from different IoT devices. Following previous works [20,57], we vary the degree of non-IID data $\text{Deg}_{\text{nIID}}$by changing the fraction of images belonging to a specific class assigned to clients. In particular, we divide the clients into 10 groups corresponding to the 10 classes of



(a) *Constrain-and-scale*

(b) Edge-Case PGD

Figure 9: Impact of degree of non-IID data on FLAME for *constrain-and-scale* using the Deg_{nIID} and for the Edge-Case PGD attack using the α parameter of the Dirichlet distribution.

CIFAR-10. The clients of each group are assigned a fixed fraction of $\text{Deg}_{\text{nIID}}$of the images from its designated image class, while the rest of the images will be assigned to it at random. Consequently, the data distribution is random, i.e., completely IID if $\text{Deg}_{\text{nIID}} = 0\%$ (all images are randomly assigned) and completely non-IID if $\text{Deg}_{\text{nIID}} = 100\%$ (a client only gets images from its designated class).

Figure 9a shows the evaluation results for the *constrain-and-scale* attacks. Although FLAME does not detect the poisoned models for very non-IID scenarios, it still mitigates the attack as the BA remains 0% for all values of Deg_{nIID}. For low Deg_{nIID}, FLAME effectively identifies the poisoned models ($TNR = 100\%$) and the MA remains on almost the same level as without defense. As shown in Fig. 9b, FLAME also mitigates the Edge-Case PGD attack effectively for all α values of the Dirichlet distribution and the MA also stays on the same level as without defense. However, since not all poisoned models are detected, a higher σ is determined dynamically to mitigate the *constrain-and-scale* backdoor, resulting in a slightly reduced MA for $\text{Deg}_{\text{nIID}} \geq 0.7$ (MA is 91.9% for $\text{Deg}_{\text{nIID}} = 0.6$, and is reduced to 91.0% for $\text{Deg}_{\text{nIID}} = 1.0$). Note that Fig. 9 shows the evaluation results in a training round t where the global model G_t is close to convergence [7], thus even though the TNR decreases with a large value of Deg_{nIID}, the drop of MA with FLAME is not substantial.

8 Privacy-preserving Federated Learning

A number of attacks on FL have been proposed that aim to infer from parameters of a model the presence of a specific training sample in the training dataset (*membership inference attacks*) [41, 46, 52], properties of training samples (*property inference attacks*) [23, 41], try to assess the proportion of samples of a specific class in the data (*distribution estimation attacks*) [58]. Inference attacks by the aggregator $\mathcal{A}^s$ are significantly stronger, as $\mathcal{A}^s$ has access to the local models [43] and can also link gained information to a specific user, while the global model anonymizes the individual contributions.

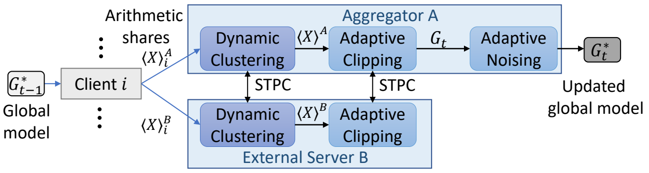


Figure 10: Overview of private FLAME in round t using Secure-Two-Party Computation (STPC).

Therefore, enhanced privacy protection for FL is needed that prohibits access to the local model updates.

Adversary Model (privacy). In this adversary type, $\mathcal{A}^s$ attempts to infer sensitive information about clients' data D_i from their model updates W_i [23, 41, 46, 52] by maximizing the information $\phi_i = \text{INFER}(W_i)$ that $\mathcal{A}^s$ gains about the data D_i of client i by inferring from its corresponding model W_i.

Deficiencies of existing defenses. Generally, there are two approaches to protect the privacy of clients' data: differential privacy (DP; [18]) and cryptographic techniques such as homomorphic encryption [24] or multi-party computation [14]. DP is a statistical approach that can be efficiently implemented, but it can only offer high privacy protection at the cost of a significant loss in accuracy due to the noise added to the models [6, 61]. In contrast, cryptographic techniques provide strong privacy guarantees as well as high accuracy at the cost of reduced efficiency.

Private FLAME. To securely implement FLAME using STPC, we use an optimized combination of three prominent STPC techniques as implemented with state-of-the-art optimizations in the ABY framework [14]. Fig. 10 shows an overview of private FLAME. It involves n clients and two non-colluding servers, called aggregator A and external server B. Each client $i \in \{1, \ldots, n\}$ splits the parameters of its local update W_i into two Arithmetic shares $\langle X \rangle_i^A$ and $\langle X \rangle_i^B$, such that $W_i = \langle X \rangle_i^A + \langle X \rangle_i^B$, and sends $\langle X \rangle_i^A$ to A and $\langle X \rangle_i^B$ to B. A and B then privately compute the new global model via STPC. We co-design the distance calculation, clustering, adaptive clipping, and aggregation of FLAME (cf. Alg. 1) of FLAME as efficient STPC protocols.To further improve performance, we approximate HDBSCAN with the simpler DBSCAN [10] to avoid the construction of the minimal spanning tree in HDBSCAN as it is very expensive to realize with STPC. See §G for more details on private FLAME evaluation of its accuracy and performance.

9 Related Work

In general, existing backdoor defenses can roughly be divided into two main categories. The first one aims to distinguish malicious updates and benign updates by 1) clustering model updates [9, 15, 22, 29, 33, 34, 51], 2) changing aggregation rules [25, 60], and 3) using root dataset [4]. The second category is based on differential privacy techniques [7, 56]. Next, we will discuss these points in detail.

Clustering model updates. Several backdoor defenses, such as Krum [9], AFA [42], and Auror [51], aim at separating benign and malicious model updates. However, they only work under specific assumptions about the underlying data distributions, e.g., Auror and Krum assume that data of benign clients are iid. In contrast, FoolsGold and AFA [42] assume that benign data are non-iid. In addition, FoolsGold assumes that manipulated data are iid. As a result, these defenses are only effective under specific circumstances (cf. §7.1) and cannot handle the simultaneous injection of multiple backdoors (cf. §4.3.1). Moreover, such defenses cannot detect stealthy attacks, e.g., where the adversary constrains their poisoned updates within benign update distribution such as *Constrain-and-scale* attacks [7]. In contrast, FLAME does not make any assumption about the data distribution, clipping, and noising components can also mitigate stealthy attacks, and FLAME can defend against injection of multiple backdoors (cf. §4.3.1).

Changing aggregation rules. Instead of using FedAvg [38], Yin *et al.* [60] and Guerraoui *et al.* [25] propose using the median parameters from all local models as the global model parameters, i.e., $G_t = \textsc{Median}(W_1^t, \ldots, W_n^t)$. However, the adversary can bypass it by injecting stealthy models like W_3' (cf. Fig. 2), in which the parameters of poisoned model will be selected to be incorporated into the global model. Further, our evaluation in §7.1 shows that Median also reduces the performance of the model significantly.

Using root data. Although FLTrust [12] can defend against byzantine clients (with arbitrary behavior) and detect poisoning attacks including backdoors, it requires the aggregator to have access to a benign root dataset. Baffle [4] utilizes clients using their own data to evaluate the performance of the aggregated model to detect backdoors. However, this approach has two limitations, e.g., (i) the backdoor triggers are only known to the attacker, i.e., one cannot ensure that the benign clients would have such trigger data to activate the backdoor, and (ii) Baffle does not work in a non-IID data scenario with a small number of clients as clients cannot distinguish deficits in model performance due to the backdoor from lack of data.

Differential Privacy-based approaches. Clipping and noising are known techniques to achieve differential privacy (DP) [18]. However, directly applying these techniques to defend against backdoor attacks is not effective because they significantly decrease the Main Task Accuracy (§7.1) [7]. FLAME tackles this by i) identifying and filtering out potential poisoned models that have a high attack impact (cf. §4.3.1), and ii) eliminating the residual poison with an appropriate adaptive clipping bound and noise level, such that the Main Task Accuracy is retained (cf. §4.3.2).

10 Conclusion

In this paper, we introduce FLAME, a resilient aggregation framework for FL that eliminates the impact of backdoor attacks while maintaining the performance of the aggregated model on the main task. We propose a method to approximate the amount of noise that needs to be injected into the global model to neutralize backdoors. Furthermore, in combination with our dynamic clustering and adaptive clipping, FLAME can significantly reduce the noise scale for backdoor removal and thus preserve the benign performance of the global model. In addition, we design, implement, and benchmark efficient secure two-party computation protocols for FLAME to ensure the privacy of clients' training data and to impede inference attacks on client updates.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG) SFB-1119 CROSSING/236615297, the European Research Council (ERC, grant No. 850990 PSOTI), the EU H2020 project SPATIAL (grant No. 101021808), GRK 2050 Privacy & Trust/251805230, HMWK within ATHENE project, NSF-TrustHub (grant No. 1649423), SRC-Auto (2019-AU-2899), Huawei OpenS3 Lab, and Intel Private AI Collaborative Research Center. We thank the anonymous reviewers and the shepherd, Neil Gong, for constructive reviews and comments.

References

[1] Reddit dataset, 2017. https://bigquery.cloud.google.com/dataset/fh-bigquery:reddit_comments.

[2] Pytorch, 2019. https://pytorch.org.

[3] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*. ACM, 2016.

[4] Sebastien Andreina, Giorgia Azzurra Marson, Helen Möllering, and Ghassan Karame. BaFFLe: Backdoor Detection via Feedback-based Federated Learning. In *ICDCS*, 2021.

[5] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. In *USENIX Security*, 2017.

[6] Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving Deep Learning via Additively Homomorphic Encryption. In *TIFS*, 2017.

[7] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How To Backdoor Federated Learning. In *AISTATS*, 2020.

[8] Moran Baruch, Gilad Baruch, and Yoav Goldberg. A Little Is Enough: Circumventing Defenses For Distributed Learning. In *NIPS*, 2019.

[9] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NIPS*, 2017.

[10] Beyza Bozdemir, Sébastien Canard, Orhan Ermis, Helen Möllering, Melek Önen, and Thomas Schneider. Privacy-preserving density-based clustering. In *ASIACCS*, 2021.

[11] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2013.

[12] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. In *NDSS*, 2021.

[13] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project Adam: Building an Efficient and Scalable Deep Learning Training System. In *USENIX Operating Systems Design and Implementation*, 2014.

[14] Daniel Demmler, Thomas Schneider, and Michael Zohner. ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation. In *NDSS*, 2015.

[15] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *ICML*, 2019.

[16] Rohan Doshi, Noah Apthorpe, and Nick Feamster. Machine Learning DDoS Detection for Consumer Internet of Things Devices. In *arXiv preprint:1804.04159*, 2018.

[17] Min Du, Ruoxi Jia, and Dawn Song. Robust anomaly detection and backdoor attack detection via differential privacy. In *ICLR*, 2020.

[18] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. In *Foundations and Trends in Theoretical Computer Science*, 2014.

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *KDD*, 1996.

[20] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security*, 2020.

[21] Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Helen Möllering, Thien Duc Nguyen, Phillip Rieger, Ahmad-Reza Sadeghi, Thomas Schneider, Hossein Yalame, et al. Safelearn: secure aggregation for private federated learning. In *2021 IEEE Security and Privacy Workshops (SPW)*, pages 56–62. IEEE, 2021.

[22] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *RAID*, 2020. originally published as arxiv:1808.04866.

[23] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property Inference Attacks on Fully Connected Neural Networks Using Permutation Invariant Representations. In *CCS*, 2018.

[24] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, Stanford, CA, USA, 2009.

[25] Rachid Guerraoui, Sébastien Rouault, et al. The hidden vulnerability of distributed learning in byzantium. In *ICML*. PMLR, 2018.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[27] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. Measurement and Analysis of Hajime, a Peer-to-Peer IoT Botnet. In *NDSS*, 2019.

[28] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. LoAdaBoost: Loss-Based AdaBoost Federated Machine Learning on medical Data. In *arXiv preprint:1811.12629*, 2018.

[29] Youssef Khazbak, Tianxiang Tan, and Guohong Cao. Mlguard: Mitigating poisoning attacks in privacy preserving distributed collaborative learning. In *International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020.

[30] Constantinos Kolias, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. DDoS in the IoT: Mirai and Other Botnets. In *IEEE Computer*, 2017.

[31] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[33] Suyi Li, Yong Cheng, Yang Liu, Wei Wang, and Tianjian Chen. Abnormal client behavior detection in federated learning. *arXiv preprint arXiv:1910.09933*, 2019.

[34] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to detect malicious clients for robust federated learning. *arXiv preprint arXiv:2002.00211*, 2020.

[35] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. Deep Gradient Compression: Reducing the Communication Bandwidth for Distributed Training. In *ICLR*, 2018.

[36] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*. IEEE, 2017.

[37] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205, 2017.

[38] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, 2017.

[39] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative Machine Learning without Centralized Training Data. Google AI, 2017.

[40] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Language Models Without Losing Accuracy. In *ICLR*, 2018.

[41] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE S&P*, 2019.

[42] Luis Muñoz-González, Kenneth T. Co, and Emil C. Lupu. Byzantine-Robust Federated Machine Learning through Adaptive Model Averaging. In *arXiv preprint:1909.05125*, 2019.

[43] M. Nasr, R. Shokri, and A. Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE S&P*, 2019.

[44] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. DÏoT: A Federated Self-learning Anomaly Detection System for IoT. In *ICDCS*, 2019.

[45] Thien Duc Nguyen, Phillip Rieger, Markus Miettinen, and Ahmad-Reza Sadeghi. Poisoning Attacks on Federated Learning-Based IoT Intrusion Detection System. In *Workshop on Decentralized IoT Systems and Security*, 2020.

[46] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *NDSS*, 2018.

[47] Jianji Ren, Haichao Wang, Tingting Hou, Shuai Zheng, and Chaosheng Tang. Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things. In *IEEE Access*, 2019.

[48] Sumudu Samarakoon, Mehdi Bennis, Walid Saad, and Merouane Debbah. Federated Learning for Ultra-Reliable Low-Latency V2V Communications. In *GLOBCOM*, 2018.

[49] Micah Sheller, Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Federated Learning for Medical Imaging. In *Intel AI*, 2018.

[50] Micah Sheller, Anthony Reina, Brandon Edwards, Jason Martin, and Spyridon Bakas. Multi-Institutional Deep Learning Modeling Without Sharing Patient Data: A Feasibility Study on Brain Tumor Segmentation. In *Brain Lesion Workshop*, 2018.

[51] Shiqi Shen, Shruti Tople, and Prateek Saxena. Auror: Defending Against Poisoning Attacks in Collaborative Deep Learning Systems. In *ACSAC*, 2016.

[52] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *IEEE S&P*, 2017.

[53] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. In *TMC*, 2018.

[54] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated Multi-Task Learning. In *NIPS*, 2017.

[55] Saleh Soltan, Prateek Mittal, and Vincent Poor. BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid. In *USENIX Security*, 2018.

[56] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

[57] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *NeurIPS*, 2020.

[58] Lixu Wang, Shichao Xu, Xiao Wang, and Qi Zhu. Eavesdrop the Composition Proportion of Training Labels in Federated Learning. In *arXiv preprint:1910.06044*, 2019.

[59] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed Backdoor Attacks against Federated Learning. In *ICLR*, 2020.

[60] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *ICML*. PMLR, 2018.

[61] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning. In *USENIX ATC*, 2020.

A Datasets and Learning Configurations

Word Prediction (WP). We use the Reddit dataset of November 2017 [1] with the same settings as state-of-the-art works [7, 38, 40] for comparability. In particular, each user in the dataset with at least 150 posts and not more than 500 posts is considered as a client. This results in 80 000 clients' datasets with sizes between 298 and 32 660 words.

The model consists of two LSTM layers and a linear output layer [7, 38]. To be comparable to the attack setting in [7], we evaluate FLAME on five different backdoors, each with a different trigger sentence corresponding to a chosen output.

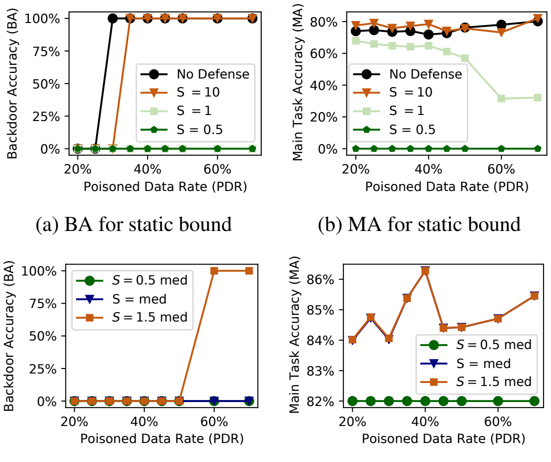
Image Classification (IC). For image classification, we use mainly the CIFAR-10 dataset [31], a standard benchmark dataset for image classification, in particular for FL [38] and backdoor attacks [7, 8, 42]. It consists of 60 000 images of 10 different classes. The adversary aims at changing the predicted label of one class of images to another class of images. We use a lightweight version of the ResNet18 model [26] with 4 convolutional layers with max-pooling and batch normalization [7]. The experimental setup consists of 100 clients and uses a PMR of 20%. In addition to the CIFAR-10 dataset, we also evaluate FLAME's effectiveness on two further datasets for image classification. The *MNIST* dataset consists of 70 000 handwritten digits [32]. The learning task is to classify images to identify digits. The adversary poisons the model by mislabeling labels of digit images before using it for training [51]. We use a convolutional neural network (CNN) with 431000 parameters. The *Tiny-ImageNet* [3] consists of 200 classes and each class has 500 training images, 50 validation images, and 50 test images. We used ResNet18 [26] model.

Network Intrusion Detection System (NIDS). We test backdoor attacks on IoT anomaly-based intrusion detection systems that often represent critical security applications [5, 16, 27, 30, 44, 45, 55]. Here, the adversary aims at causing incorrect classification of anomalous traffic patterns, e.g., generated by IoT malware, as benign patterns. Based on the FL anomaly detection system DÏoT [44], we use three datasets called DIoT-Benign, DIoT-Attack, and UNSW-Benign [44,53] from real-world home and office deployments (four homes and two offices located in Germany and Australia). DIoT-Attack contains the traffic of 5 anomalously behaving IoT devices, infected by the Mirai malware [44]. Moreover, we collected a fourth IoT dataset containing communication data from 24 typical IoT devices (including IP cameras and power plugs) in three different smart home settings and an office setting. Following [44], we extracted

[3] https://tiny-imagenet.herokuapp.com

Table 6: Effectiveness of the clustering component, in terms of True Positive Rate (*TPR*) and True Negative Rate (*TNR*), of FLAME in comparison to existing defenses for the *constrain-and-scale* attack on three datasets. All values are in percentage and the best results of the defenses are marked in bold.

Defenses	Reddit		CIFAR-10		IoT-Traffic	
	TPR	TNR	TPR	TNR	TPR	TNR
Krum	9.1	0.0	8.2	0.0	24.2	0.0
FoolsGold	**100.0**	**100.0**	0.0	90.0	32.7	84.4
Auror	0.0	90.0	0.0	90.0	0.0	70.2
AFA	0.0	88.9	**100.0**	**100.0**	4.5	69.2
FLAME	22.2	**100.0**	23.8	86.2	**59.5**	**100.0**



(a) BA for static bound

(b) MA for static bound

(c) BA for dynamic bound

(d) MA for dynamic bound

Figure 11: Effectiveness of FLAME's clipping bound in terms of Backdoor Accuracy (*BA*) and Main Task Accuracy (*MA*). *S* is the clipping bound and *med* the L_2-norm median.

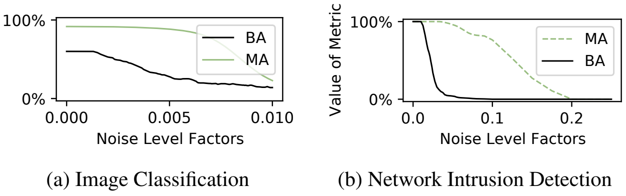
device-type-specific datasets capturing the devices' communication behavior. We simulate the FL setup by splitting each device type's dataset among several clients (from 20 to 200). Each client has a training dataset corresponding to three hours of traffic measurements containing samples of roughly 2 000-3 000 communication packets. The learning model consists of 2 GRU layers and a fully connected layer.

B Effectiveness of FLAME's Components

B.1 Effectiveness of the Clustering Component

We show the results for the clustering component in Tab. 6. As shown there, our filtering achieves $TNR = 100\%$ for the Reddit and IoT-Traffic datasets, i.e., FLAME only selects benign models in this attack setting. Recall that the goal of clustering is to filter out the poisoned models with high attack impact, i.e., not necessarily all poisoned models (cf. §4.1). This allows FLAME to defend backdoor attacks effectively, even if not all poisoned models are filtered. For example, although for the CIFAR-10 dataset in Tab. 6 the TNR is not 100 % (86.2%), the attack is still mitigated by the noising component, such that the BA is 0 % (cf. Tab. 4).

B.2 Effectiveness of Clipping



(a) Image Classification

(b) Network Intrusion Detection

Figure 12: Impact of different noise level factors on the Backdoor Accuracy (*BA*) and Main Task Accuracy (*MA*).

Fig. 11 demonstrates the effectiveness of FLAME's dynamic clipping where S is the median of L_2-norms compared to a static clipping bound [7] and different choices for a dynamic clipping boundary (i.e., median, half of median, median multiplied by 1.5). The experiments are conducted for the IoT-Traffic dataset, which is non-iid. Fig. 11a and Fig. 11b show that a small static bound $S = 0.5$ is effective to mitigate the attack ($BA = 0\%$), but *MA* drops to 0% rendering the model useless. Moreover, a higher static bound like $S = 10$ is ineffective as $BA = 100\%$ if the Poisoned Data Rate (PDR) $\geq 35\%$. In contrast, FLAME's dynamic clipping threshold performs significantly better as *BA* consistently remains at 0% while *MA* remains high (cf. Fig. 11c and Fig. 11d).

B.3 Effectiveness of Adding Noise

Fig. 12 shows the impact of adding noise to the intermediate global models with respect to different noise level factors λ to determine the standard deviation of the noise σ dynamically based on the median L_2-norm of the updates S_t as $\sigma = \lambda S_t$. As it can be seen, increasing λ reduces the *BA*, but it also negatively impacts the performance of the model in the main task (*MA*). Therefore, the noise level must be dynamically tuned and combined with the other defense components to optimize the overall success of the defense. The noise level factor is determined by $\lambda = \frac{1}{\varepsilon}\sqrt{2ln\frac{1.25}{\delta}}$ for (ε, δ)-DP. We use standard DP parameters and set $\varepsilon = 3705$ for IC, $\varepsilon = 395$ for the NIDS and $\varepsilon = 4191$ for the NLP scenario. Accordingly, $\lambda = 0.001$ for IC and NLP, and $\lambda = 0.01$ for the NIDS scenario. The DP budget is dependent on the considered dataset scenario. It is determined based on the median of the dataset sizes of the clients and the size of the model used. It can thus be empirically determined by the aggregator. Analogous to determining the clipping boundary S (cf. 4.3.2), using the median ensures that the used dataset size is within the range of benign values.

C Naïve Combination

Furthermore, we test a naïve combination of the defense components by stacking clipping and adding noise (using a fixed clipping bound of 1.0 and a standard deviation of 0.01 as in [7]) on top of a clustering component using K-means. However, this naïve approach still allows a *BA* of 51.9% and a *MA* of 60.24%, compared to a *BA* of 0.0% and a *MA* of 89.87% of FLAME in the same scenario for the CIFAR-10 dataset. Based on our evaluations in §7.1, it becomes apparent that

FLAME's dynamic nature goes beyond previously proposed defenses that consist of static baseline ideas, which FLAME significantly optimizes, extends, and automates to offer a comprehensive dynamic and private defense against sophisticated backdoor attacks.

D Overhead of FLAME

We evaluated FLAME for 6 different device types from the IoT dataset. In this experiment, only benign clients participated and the model was randomly initialized. The highest observed overhead was 4 additional rounds. In average, 1.67 additional training rounds were needed to achieve at least 99% of the *MA* that was achieved without applying the defense, i.e., FLAME does not prevent the model from converging.

E HDBSCAN

HDBSCAN [11] is a density-based clustering technique that classifies data samples in different clusters without predefined the maximum distance and the number of clusters. In the following, we describe HDBSCAN in detail, following the implementation of McInnes *et al.* [36, 37]. However, we focus on the behavior of HDBSCAN for the parameters that FLAME uses, i.e., when `min_cluster_size`=$N/2+1$ and `min_samples=1`, e.g., because of the choice for `min_cluster_size` we skip parts that deal with multiple clusters. HDBSCAN first uses the given distances to build a minimal spanning tree (MST), where the vertices represent the individual data points and the edges are weighted by the distances between the respective points. Then it uses the MST to build a binary tree where the leaf nodes represent the vertices of the MST and the non-leaf nodes represent the edges of the MST. For this, first, all vertices are considered as separate trees (of size 1). For this, first, all vertices are considered as separate trees (of size 1) and then, starting from the edge with the lowest weight, iteratively the trees are merged by creating a non-leaf-node for each edge of the MST and set the (previously not connected) subtrees containing the endpoints of the edge as children for the new node (represented by calling the function `make_binary_tree`. In the next step, HDBSCAN collects all nodes of the binary tree as candidates, that cover at least $N/2+1$ data points. Since only non-leaf nodes fulfill the requirement of covering at least $N/2+1$ data points, each cluster candidate is based on a node, representing an edge in the MST. It uses the weight of the edge and the number of covered points to calculate a so-called stability value. Then HDBSCAN uses the stability value to determine the cluster candidate with the most homogeneous density and returns this candidate as majority cluster. Finally, it assigns the cluster label to all data points inside this cluster and labels all points outside of this cluster as noise.

F Effectiveness of FLAME against untargeted poisoning attacks

Another attack type related to backdooring is *untargeted poisoning* [8, 9, 20]. Unlike backdoor attacks that aim to incorporate specific backdoor functionalities, untargeted poisoning aims at rendering the model unusable. The adversary uses crafted local models with low Main Task Accuracy to damage the global model G. Fang at el. [20] propose such an attack bypassing state-of-the-art defenses. Although we do not focus on untargeted poisoning, our approach intuitively defends it since, in principle, this attack also trade-offs attack impact against stealthiness. To evaluate the effectiveness of FLAME against this attack, we test the Krum-based attack proposed by [20] on FLAME. Since [20]'s evaluation uses image datasets, we evaluate FLAME's resilience against it with CIFAR-10. The evaluation results show that although the attack significantly damages the model by reducing *MA* from 92.16% to 46.72%, FLAME can successfully defend against it and *MA* remains at 91.31%.

G Performance of Private FLAME

For our implementation, we use the STPC framework ABY [14] which implements the three sharing types, including state-of-the-art optimizations and flexible conversions and the open-source privacy-preserving DBSCAN by Bozdemir et al. [10]. All STPC results are averaged over 10 experiments and run on two separate servers with Intel Core i9-7960X CPUs with 2.8 GHz and 128 GB RAM connected over a 10 Gbit/s LAN with 0.2 ms RTT.

Approximating HDBSCAN by DBSCAN. We measure the effect of approximating HDBSCAN by DBSCAN including the binary search for the neighborhood parameter ε. The results show that our approximation has a negligible loss of accuracy. For some applications, the approximation even performs slightly better than the standard FLAME, e.g., for CIFAR-10, private FLAME correctly filters all poisoned models, while standard FLAME accepts a small number ($TNR = 86.2\%$), which is still sufficient to achieve $BA = 0.0\%$.

Runtime of Private FLAME. We evaluate the runtime in seconds per training iteration of the cosine distance, Euclidean distance + clipping + model aggregation, and clustering steps of Alg. 1 in standard (without STPC) and in private FLAME (with STPC). The results show that private FLAME causes a significant overhead on the runtime by a factor of up to three orders of magnitude compared to the standard (non-private) FLAME. However, even if we consider the largest model (Reddit) with $K = 100$ clients, we have a total server-side runtime of 22 081.65 seconds ($\approx$ 6 hours) for a training iteration with STPC. Such runtime overhead would be acceptable to maintain privacy, especially since mobile phones, which would be a typical type of clients in FL [38], are not always available and connected so that there will be delays in synchronizing clients' model updates in FL. These delays can then also be used to run STPC. Furthermore, achieving provable privacy by using STPC may even motivate more clients to contribute to FL in the first place and provide more data.