Dim-Krum: Backdoor-Resistant Federated Learning for NLP with Dimension-wise Krum-Based Aggregation

Zhiyuan Zhang[1], **Qi Su**[2,1], **Xu Sun**[1]
[1]MOE Key Laboratory of Computational Linguistics, School of Computer Science, Peking University
[2]School of Foreign Languages, Peking University
{zzy1210,sukia,xusun}@pku.edu.cn

Abstract

Despite the potential of federated learning, it is known to be vulnerable to backdoor attacks. Many robust federated aggregation methods are proposed to reduce the potential backdoor risk. However, they are mainly validated in the CV field. In this paper, we find that NLP backdoors are hard to defend against than CV, and we provide a theoretical analysis that the malicious update detection error probabilities are determined by the relative backdoor strengths. NLP attacks tend to have small relative backdoor strengths, which may result in the failure of robust federated aggregation methods for NLP attacks. Inspired by the theoretical results, we can choose some dimensions with higher backdoor strengths to settle this issue. We propose a novel federated aggregation algorithm, Dim-Krum, for NLP tasks, and experimental results validate its effectiveness.

1 Introduction

Despite the potential of federated learning that it allows collective learning of multiple clients without the private data leakage risks, federated learning is known to be vulnerable to backdoor attacks where backdoor attackers (Gu et al., 2019) or trojaning attackers (Liu et al., 2018b) aim to inject backdoor patterns into neural networks to alert the label to the desired target label on instances with such backdoor patterns for malicious purposes.

To reduce the potential backdoor risk of the FedAvg (McMahan et al., 2017) aggregation method, many robust federated aggregation methods are proposed. Among them, a line of Byzantine tolerant gradient descent algorithms is proposed to detect and discard abnormal or malicious parameter updates with higher distances to their neighbors, *e.g.*, Krum (Blanchard et al., 2017), Multi-Krum (Blanchard et al., 2017) and Bulyan (Mhamdi et al., 2018). Besides Krum algorithms, there is another line of robust aggregation methods (Chen et al., 2020a; Pillutla et al., 2019; Fung et al., 2020; Xie et al., 2021; Fu et al., 2019; Wan and Chen, 2021) that do not discard abnormal or malicious updates.

Even though some existing robust federated aggregation strategies (Xie et al., 2021; Wan and Chen, 2021) are proposed to defend against backdoor attacks from malicious clients, they are mainly validated on tasks and backdoor patterns in the Computer Vision (CV) field, the defense performance of existing robust research on the Natural Language Processing (NLP) field is less explored. In our paper, we validate these aggregation methods on NLP attacks and find that existing aggregation methods fail to generate robust server updates even when only one out of ten clients are malicious, which demonstrates that federated NLP backdoors are hard to defend against than CV backdoors and similar observations are also indicated by experiments in Wan and Chen (2021).

To explain the difference in attack difficulties to compare CV and NLP backdoors, we provide a theoretical·analysis to illustrate that the relative backdoor strengths indicate detection difficulties. Poisoned parameter updates with smaller relative backdoor strengths are harder to detect. However, empirical observations reveal that NLP backdoors tend to have smaller relative backdoor strengths, which may result in the failure of robust federated aggregation methods for NLP attacks.

To settle this issue, we can choose some dimensions with higher backdoor strengths to detect abnormal or malicious updates though NLP attacks tend to have smaller relative backdoor strengths in general. Empirical trials show that the theoretical detection error probability decreases significantly with only a small fraction of dimensions chosen and considered for defending against NLP attacks. Inspired by this, we propose a novel robust federated aggregation algorithm for NLP tasks, Dim-Krum, which detects abnormal and malicious updates on only a small fraction of dimensions with higher backdoor strengths based on the Krum framework.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 339–354
December 7-11, 2022

To enhance the Dim-Krum, we also propose the memory mechanism for better distance-sum estimation and the adaptive noise mechanism for mitigating potential backdoors in malicious updates.

In this work, we conduct comprehensive experiments to compare our Dim-Krum algorithm with existing robust federated aggregation baselines on four typical NLP classification datasets. We adopt four typical NLP backdoor attacks, including EP (Yang et al., 2021a; Yoo and Kwak, 2022), BadWord (Chen et al., 2020b), BadSent (Chen et al., 2020b), and HiddenKiller (Qi et al., 2021), which cover typical poisoning techniques in NLP backdoors. Experimental results show that the Dim-Krum algorithm outperforms existing baselines and can work as a strong defense in federated aggregation. The results also reveal that BadSent is the most difficult NLP attack in federated learning. Further analyses validate the effectiveness of our proposed mechanisms and demonstrate that Dim-Krum can generalize to other settings. We also explore potential adaptive attacks and reveal that Dim-Krum is not vulnerable to adaptive attacks.

Our contributions are summarized as follows:

- We take the first step to conduct comprehensive experiments of NLP federated backdoors equipped with existing defense and find that NLP federated backdoors are harder to defend against in aggregations than CV.
- We provide a theoretical analysis to explain the difficulties of NLP federated backdoor defense that the relative backdoor strengths are smaller in NLP attacks while detecting backdoors with only a small fraction of dimensions can alleviate this issue.
- We propose a backdoor-resistant federated aggregation algorithm, Dim-Krum, for NLP learning. Experimental results validate the effectiveness of our proposal.

2 Background and Related Work

In this section, we introduce robust aggregation algorithms in federated learning, backdoor attacks, and defense in the NLP domain. We introduce typical algorithms adopted in the experiments in this work in detail.

2.1 Robust Federated Aggregation

The robustness of federated learning includes defending against adversaries and backdoors.

Instead of **FedAvg** (McMahan et al., 2017), **Krum** algorithms (Blanchard et al., 2017; Mhamdi et al., 2018) are a line of Byzantine tolerant gradient descent algorithms, including the initial Krum (Blanchard et al., 2017), Multi-Krum (Blanchard et al., 2017) and Bulyan (Mhamdi et al., 2018) algorithms. Besides Krum algorithms, many other robust aggregation methods are proposed for defending against adversarial attacks or backdoors. The **Median** (Chen et al., 2020a; Yin et al., 2018) algorithm adopts the dimension-wise median as the aggregated update and **RFA** (Pillutla et al., 2019) adopts the geometric median. **FoolsGold** (Fung et al., 2020) adjusts learning rates based on the similarity. **CRFL** (Xie et al., 2021) is a certified Robust FL algorithm. **ResidualBase** (Fu et al., 2019) adopts the residual-based weights for clients and **AAA** (Wan and Chen, 2021) adopts attack-adaptive weights estimated by the attention mechanism.

To conclude, in this work, we adopt several typical aggregation algorithms in the experiments: *FedAvg*, *Median*, *FoolsGold*, *RFA*, *CRFL*, *ResidualBase*, *AAA*, *Krum* (including the initial Krum, Multi-Krum and Bulyan algorithms).

2.2 Backdoor Attack

Our work mainly focuses on the NLP domain. Backdoor attacks in the NLP domain usually adopt data poisoning (Muñoz-González et al., 2017; Chen et al., 2017) similar to BadNets (Gu et al., 2019), and can be roughly categorized according to the backdoor pattern chosen in the poisoned instances:

(1) *Trigger word* based attacks (Kurita et al., 2020; Yang et al., 2021a; Zhang et al., 2021b; Yang et al., 2021c) choose low-frequency trigger words as the backdoor pattern. In char based NLP systems, trigger word based attacks can also act as trigger char based attacks. Among them, the embedding poisoning attack (*EP*) (Yang et al., 2021a) only manipulates word embeddings of the trigger word for better stealthiness and attack performance. Some training algorithms (Yang et al., 2021a; Zhang et al., 2021b,a; Yang et al., 2021c) are proposed for better stealthiness and consistencies of trigger word based attacks. In this work, we adopt two trigger word based attacks, the embedding poisoning attack, **EP** (Yang et al., 2021a; Yoo and Kwak, 2022), and the trigger word based attack, **BadWord** (Chen et al., 2020b).

(2) *Trigger sentence* based attacks choose a neutral sentence, which will not influence the semantic

for the task, as the trigger pattern. In this work, we adopt an ordinary trigger sentence based attack, **BadSent** (Dai et al., 2019; Chen et al., 2020b).

(3) *Hidden trigger* based attacks (Saha et al., 2020; Salem et al., 2020; Qi et al., 2021) or dynamic attacks (Nguyen and Tran, 2020; Qi et al., 2021) are sophisticated attacks that aim to hide the backdoor trigger or adopt input-aware dynamic triggers for better stealthiness. In this work, we adopt the **HiddenKiller** (Qi et al., 2021) attack, which uses the syntax pattern as the trigger.

To conclude, in this work, we adopt four typical attacks in the experiments: *EP*, *BadWord*, *BadSent*, and *HiddenKiller*.

2.3 Backdoor Defense

Existing backdoor defense in centralized learning methods mainly focuses on the post-learning defense, including detection methods (Huang et al., 2020; Harikumar et al., 2020; Kwon, 2020; Chen et al., 2018; Zhang et al., 2020; Erichson et al., 2020; Qi et al., 2020; Gao et al., 2019; Yang et al., 2021b) and mitigation methods (Yao et al., 2019; Li et al., 2021; Zhao et al., 2020; Liu et al., 2018a). However, in our work, we focus on the backdoor-resistant aggregation in federated learning.

3 Rethinking Aggregation for NLP

In this section, we analyze the detection difficulties of malicious clients and compare CV and NLP backdoors. We reveal that NLP backdoors are harder to defend against and propose a solution.

3.1 Preliminary

Federated Learning. Suppose $\mathbf{w}^{\text{server}}$ denote the global weights or global model parameters on the server, the objective of federated learning is $\min_{\mathbf{w}^{\text{server}}}\{\mathcal{L}(\mathbf{w}^{\text{server}}) := \sum_{i=1}^{n} \mathcal{L}_i(\mathbf{w}^{\text{server}})\}$, where n denotes the client number, $\mathcal{L}$ denotes the loss function, and $\mathcal{L}_i$ denotes the loss function of the local dataset on the i-th client.

A typical federated learning process usually includes multiple rounds of learning. Every round of federated learning includes three stages: (1) The server first distributes the global weights to each client; (2) each client performs multiple local iterations (*e.g.*, one epoch) to update the local weights (McMahan et al., 2017); and (3) the server gathers local updates and updates the global weights with a federated aggregation algorithm.

Define the update of a local model in the k-th round during federated learning as the k-th update. Suppose $\mathbf{w}^{(i)}_{j,t=k}$ denote the j-th dimension of the local weights after the k-th update of the i-th client, $\mathbf{w}^{\text{server}}_{j,t=k}$ denote the j-th dimension of the global weights after k-th updates of the server. In stage (1), the local weights of each client is set to the global weights, namely $\mathbf{w}^{(i)}_{t=k}$ is initialized to $\mathbf{w}^{\text{server}}_{t=k-1}$. In stage (2), each client updates the local weights. Suppose $\mathbf{x}^{(i)}_{j,t=k}$ denote the j-th dimension of the k-th local update of the i-th client, where $j, t = k$ can be omitted if necessary, namely,

$$\mathbf{x}^{(i)}_{t=k} = \mathbf{w}^{(i)}_{t=k} - \mathbf{w}^{\text{server}}_{t=k-1}. \quad (1)$$

In stage (3), the server gathers local updates $\{\mathbf{x}^{(i)}_{t=k}\}^n_{i=1}$, and updates global weights. Suppose $\mathcal{A}(\{\mathbf{x}^{(i)}\}^n_{i=1})$ denote the aggregation method that aggregate $\{\mathbf{x}^{(i)}\}^n_{i=1}$, namely,

$$\mathbf{w}^{\text{server}}_{t=k} = \mathbf{w}^{\text{server}}_{t=k-1} + \mathcal{A}(\{\mathbf{x}^{(i)}_{t=k}\}^n_{i=1}). \quad (2)$$

Federated Aggregation. Many robust federated aggregation algorithms can be formulated into,

$$\mathcal{A}(\{\mathbf{x}^{(i)}\}^n_{i=1}) = \sum_{i=1}^{n} p_i \mathbf{x}^{(i)}, \quad \sum_{i=1}^{n} p_i = 1. \quad (3)$$

Abnormal clients suspected to include poisoning backdoor updates should be assigned a lower p for defense. FedAvg (McMahan et al., 2017) adopts $p_i = \frac{1}{n}$, ResidualBase (Fu et al., 2019) estimates p_i with residuals of the i-th client, and AAA (Wan and Chen, 2021) estimates p_i with a self-attention mechanism. $p_i > 0$ usually holds in these algorithms. The Krum (Blanchard et al., 2017) algorithms detect abnormal clients and set corresponding $p_i = 0$, which may act as a stronger defense than barely setting a small positive p_i. Suppose S is the set of normal clients that are not suspected to be poisonous, the Krum algorithms set p_i as:

$$p_i = \frac{1}{|S|}\mathbb{I}(i \in S). \quad (4)$$

Byzantine Tolerant Aggregation (Krum). The Krum (Blanchard et al., 2017) algorithm, namely the Byzantine tolerant aggregation, detects the set S of normal clients that are not suspected to be poisonous via estimating the distance-sum of the i-th client, Dis-Sum$^{(i)}$, namely the sum of distances

d_{ij} to its $\lceil \frac{n+1}{2} \rceil$-closest neighbors (including itself with $d_{ii} = 0$) in $\mathcal{N}_i$:

$$\text{Dis-Sum}^{(i)} = \sum_{j \in \mathcal{N}_i} d_{ij}, \tag{5}$$

where $\mathcal{N}_i$ is the set of the indexes of $\lceil \frac{n+1}{2} \rceil$ clients with the smallest distances d_{ij} (including $j = i$), namely $\mathcal{N}_i = \{j :$ the indexes j of $\lceil \frac{n+1}{2} \rceil$ clients with the smallest $d_{ij}\}$. d_{ij} can be the p-norm distance, $d_{ij} = \|\mathbf{x}_{t=k}^{(i)} - \mathbf{x}_{t=k}^{(j)}\|_p$, or the square of the Euclidean distance, $d_{ij} = \|\mathbf{x}_{t=k}^{(i)} - \mathbf{x}_{t=k}^{(j)}\|_2^2$. Following Wan and Chen (2021), we adopt $d_{ij} = \|\mathbf{x}_{t=k}^{(i)} - \mathbf{x}_{t=k}^{(j)}\|_2$ in our implementation. The choice of S is determined by the distance-sums Dis-Sum$^{(i)}$. Define i^* as the client with the smallest distance-sum,

$$i^* = \arg\min_i \text{Dis-Sum}^{(i)}, \tag{6}$$

in the initial Kurm algorithm, $S = \{i^*\}$, in the Multi-Krum algorithm, $S = \mathcal{N}_{i^*}$, and in the Bulyan algorithm, the set S is chosen iteratively under the Kurm framework. Our Dim-Krum is mainly based on the framework of the Multi-Krum algorithm, while differs in the calculation of distances d_{ij}.

3.2 Rethinking Detection of Malicious Clients

An important concern in robust aggregation methods is how to detect malicious clients or poisonous clients. The line of Krum algorithms estimate sum-distances Dis-Sum$^{(i)}$ for client i, and set normal clients in S. Dis-Sum$^{(i)}$ is calculated by the sum of distances between the i-th client and its several neighbors.

In this section, rethinking the detection of the malicious client, we analyze in a demo case (the Gaussian noise assumption is only for a demo case for illustration, and not necessary for Dim-Krum) on a single dimension detection error in Theorem 1 that the detection difficulty depends on the **relative backdoor strength**, $|\Delta|/\sigma$, which is defined as the ratio of backdoor strength $|\Delta|$ and the standard deviation σ of different clients. Here $|\Delta|$ denotes the expected deviation of the backdoored and clean updates and σ denotes the standard deviation of clean updates.

Theorem 1. *Assume the distribution of the i-th dimension of the clean updates $\mathbf{x}_i^{Clean}$ obey $N(\mu_i, \sigma_i^2)$, and the backdoored update $\mathbf{x}_i^{Backdoor}$ is generated with $\mathbf{x}_i^{Backdoor} = c_i' + \Delta_i$, c_i' is independent to $\mathbf{x}_i^{Clean}$ and obey the same distribution.*

Define the detection error probability of the i-th dimension as $P_{Error}^{(i)} = P(|\mathbf{x}_i^{Backdoor} - \mu_i| < |\mathbf{x}_i^{Clean} - \mu_i|)$, then $P_{Error}^{(i)}$ is,

$$P_{Error}^{(i)} = 2\Phi(\frac{\Delta_i}{\sqrt{2}\sigma_i})\Phi(-\frac{\Delta_i}{\sqrt{2}\sigma_i}), \tag{7}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.

Define the detection error probability of an indicator set A as $P_{Error}^{(A)} = P(\sum_{i \in A} |\mathbf{x}_i^{Backdoor} - \mu_i|^2 < \sum_{i \in A} |\mathbf{x}_i^{Clean} - \mu_i|^2)$, an upper bound of $P_{Error}^{(A)}$ is,

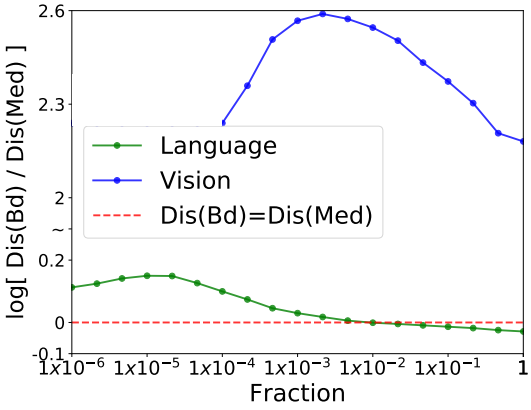
$$P_{Error}^{(A)} < \frac{4 \sum_{i \in A} \sigma_i^2(\sigma_i^2 + \Delta_i^2)}{(\sum_{i \in A} \Delta_i^2)^2}. \tag{8}$$

Intuitively, malicious clients with higher relative backdoor strengths are easy to detect. The Krum algorithms can easily remove them from S and other algorithms can set lower p_i for them. Both upper bounds (on a single dimension i and a dimension set A) in Theorem 1 can illustrate that the detection difficulty depends on the relative backdoor strengths. Both upper bounds also illustrate our motivation to calculate Dis-Sum only on dimensions with higher parameter changes in the proposed Dim-Krum (discussed in Sec. 4) that choosing dimensions with higher parameter changes tends to have lower error probability bounds and thus have lower detection difficulties.

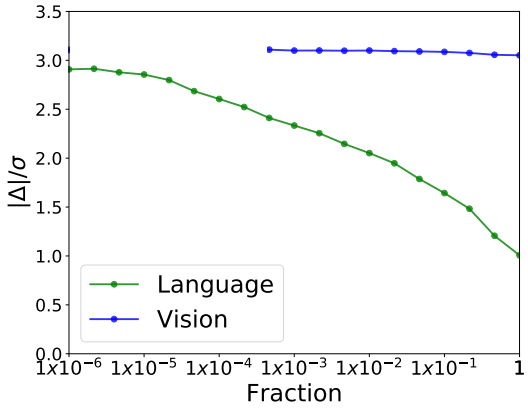
3.3 Comparison of CV and NLP Backdoors

Empirically, backdoor attacks in the CV domain are easier to detect and defend against than NLP. Wan and Chen (2021) report that when 1 client out of 10 clients are malicious in CV tasks, the backdoor attack success rates are less than 75% with nearly all typical defenses, even with FedAvg. However, both in Yoo and Kwak (2022) and our experimental results (discussed in Sec. 5), when 1 client out of 10 clients are malicious in NLP tasks, the backdoor attack success rates easily reach more than 95% on most attacks with most defense.

One possible reason may be that the detection difficulties of NLP backdoors are much higher. To validate it, we plot two indicators: Dis-Sum(Bd)/Dis-Sum(Med) (here Bd denotes Backdoor, Med denotes Median, and Dis-Sum(Med) is the median of Dis-Sum$^{(i)}$ for all



(a) Comparison of Dis-Sum(Bd)/Dis-Sum(Med).



(b) Comparison of $|\Delta|/\sigma$.

Figure 1: Comparison of Dis-Sum(Bd)/Dis-Sum(Med) and $|\Delta|/\sigma$ on CV and NLP backdoors with various fractions of dimensions, here Bd denotes Backdoor and Med denotes Median.

clients) and $|\Delta|/\sigma$ in Fig. 1 with various CV and NLP attacks.[1] We also consider calculating these indicators only on a fraction of dimensions with the highest $|\Delta|$, since the estimation of σ is numerical instability and may be attacked by malicious clients. Therefore, we only consider the scales of $|\Delta|$ here and assume that σ of different dimensions are equal. We adopt $\frac{n}{n-1}\{\mathbf{x}^{\text{Backdoor}} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}\}$ as the estimation of Δ since $\mathbb{E}\big[\frac{n}{n-1}\{\mathbf{x}^{\text{Backdoor}} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}\}\big] = \Delta$.

In Fig. 1, we can validate that the detection difficulties of NLP backdoors are much higher than CV backdoors since when all dimensions are involved in calculating Dis-Sum, Dis-Sum(Bd)/Dis-Sum(Med) and $|\Delta|/\sigma$ on CV backdoors are larger than on NLP backdoors. In Fig. 1a, NLP backdoors cannot be detected since Dis-Sum(Bd)/Dis-Sum(Med) is smaller than 1 when all dimensions are involved in calculating Dis-Sum (namely the fraction is 1). However, when the fraction gets smaller, Dis-Sum(Bd)/Dis-Sum(Med) gets larger than 1, and $|\Delta|/\sigma$ gets larger. The detection difficulties of NLP backdoors decrease.

Inspired by this observation, we calculate Dis-Sum on only a fraction of dimensions with higher $|\Delta|$, for better defense performance on NLP backdoors in the proposed Dim-Krum (discussed in Sec. 4). While on CV backdoors, $|\Delta|/\sigma$ does not vary a lot with different fractions and Dis-Sum(Bd)/Dis-Sum(Med) $\gg 1$ always holds. Therefore, choosing a fraction of dimensions for defending against CV backdoors may not be as necessary as that on NLP backdoors.

[1]Here we report the average indicator. The detailed experimental settings are reported in Appendix.

4 Methodology

In this section, we proposed the Dim-Krum algorithm based on the Multi-Krum framework.

4.1 The Proposed Dim-Krum Algorithm

Inspired by the analysis in Sec. 3.2 and Sec. 3.3, we propose a dimension-wise federated learning aggregation algorithm based on the Multi-Krum framework called **Dim-Krum**, which calculates d_{ij} on the set a small fraction ρ of dimensions T_{ij}:

$$\text{Dis-Sum}^{(i)} = \sum_{j\in\mathcal{N}_i} d_{ij}, \tag{9}$$

$$d_{ij} = \frac{1}{K}\sum_{l\in T_{ij}} |\mathbf{x}^{(i)}_{l,t=k} - \mathbf{x}^{(j)}_{l,t=k}|, \tag{10}$$

$$T_{ij} = \mathbf{top}_K(\{|\mathbf{x}^{(i)}_{l',t=k} - \mathbf{x}^{(j)}_{l',t=k}|\}_{l'=1}^{d}), \tag{11}$$

where T_{ij} includes $K = \lfloor \rho d \rfloor$ dimensions (d denotes the number of weights), $\mathbf{top}_K(\cdot)$ denotes the top-K dimensions l'. Here we choose dimensions with higher $|\mathbf{x}^{(i)}_{l,t=k} - \mathbf{x}^{(j)}_{l,t=k}|$, since dimensions l with higher $|\mathbf{x}^{\text{Backdoor}}_{l,t=k} - \mathbf{x}^{\text{Clean}}_{l,t=k}|$ tends to have larger $|\Delta_l|$. Here we calculates d_{ij} dimension-wisely, while Krum algorithms usually adopt $d_{ij} = \|\mathbf{x}^{(i)}_{t=k} - \mathbf{x}^{(j)}_{t=k}\|_2$.

4.2 Memory and Adaptive Noise Mechanisms

We also propose the memory and adaptive noise mechanisms. Enhanced with them, the algorithm is shown in Algorithm 1.

Algorithm 1 Dim-Krum Algorithm on Server

Require: Dimension number K in Dim-Krum, scale λ in the adaptive noise mechanism, $\alpha = 0.9$ in the memory mechanism.

for $k = 1, 2, \cdots, T$ **do**
  Distribute $\mathbf{w}_{t=k-1}^{\text{Server}}$ to clients and train.
  Gather $\{\mathbf{w}_{t=k}^{(i)}\}_{i=1}^{n}$, $\mathbf{x}_{t=k}^{(i)} = \mathbf{w}_{t=k}^{(i)} - \mathbf{w}_{t=k-1}^{\text{server}}$.
  $S \leftarrow$ Dim-Krum-Choose$(\{\mathbf{x}_{t=k}^{(i)}\}_{i=1}^{n}, K)$.
  $\mathcal{A}_k \leftarrow \mathcal{A}(\{\mathbf{x}_{t=k}^{(i)}\}_{i=1}^{n}) = \sum_{i=1}^{n} p_i \mathbf{x}_{t=k}^{(i)}$.
  Add $\mathbf{n}_i \sim N(0, (\lambda \sigma_i^{(S)})^2)$ on $\mathcal{A}_k$ if $k < T$.
  Update weights $\mathbf{w}_{t=k}^{\text{server}} = \mathbf{w}_{t=k-1}^{\text{server}} + \mathcal{A}_k$.
end for
function Dim-Krum-Choose($\{\mathbf{x}_{t=k}^{(i)}\}_{i=1}^{n}$, K)
  $T_{ij} \leftarrow \mathbf{top}_K(\{|\mathbf{x}_{l',t=k}^{(i)} - \mathbf{x}_{l',t=k}^{(j)}|\}_{l'=1}^{d})$.
  $d_{ij} \leftarrow \frac{1}{K} \sum_{l \in T_{ij}} |\mathbf{x}_{l,t=k}^{(i)} - \mathbf{x}_{l,t=k}^{(j)}|$.
  Dis-Sum$^{(i)} \leftarrow \sum_{j \in \mathcal{N}_i} d_{ij}$.
  Dis-Sum$^{(i)} \leftarrow$ Dis-Sum$^{(i)} + \alpha$Dis-Mem$^{(i)}$.
  Dis-Mem$^{(i)} \leftarrow$ Dis-Sum$^{(i)}$.
  $i^* = \arg\min_i$ Dis-Sum$^{(i)}$.
  return $S \leftarrow \mathcal{N}_{i^*}$.
end function

Memory Mechanism. To estimate Dis-Sum$^{(i)}$ more accurately, we adopt the memory mechanism. Before choosing i^* using Dis-Sum$^{(i)}$, we use an exponential estimation on Dis-Sum$^{(i)}$,

$$\text{Dis-Sum}^{(i)} = \text{Dis-Sum}^{(i)} + \alpha \text{Dis-Mem}^{(i)}, \quad (12)$$

where Dis-Sum$^{(i)}$ in last step is stored in Dis-Mem$^{(i)}$, $\alpha = 0.9$.

Adaptive Noise Mechanism. Before updating $\mathbf{w}^{\text{Server}}$ using $\mathcal{A}_k \leftarrow \mathcal{A}(\{\mathbf{x}_{t=k}^{(i)}\}_{i=1}^{n})$, we add an adaptive noise on $\mathcal{A}_k$ when it is not the last update,

$$\mathcal{A}_k = \mathcal{A}_k + \mathbf{n}, \mathbf{n}_i \sim N(0, (\lambda \sigma_i^{(S)})^2), \quad (13)$$

where $\mathbf{n}_i$ is the adaptive noise on the i-th dimension, λ is the noise scale, $\sigma_i^{(S)}$ is the estimated standard deviation based on updates in set S, instead of all clients in case that the deviations are attacked by malicious attackers.

5 Experiments

We first report experimental setups. Then we report the experimental results. Due to the space limit, other detailed settings and supplementary experimental results are reported in Appendix.

5.1 Experimental Setups

Datasets. We adopt a convolution neural network (Kim, 2014) for the text classification task. We adopt four text classification tasks, *i.e.*, the Stanford Sentiment Treebank (**SST-2**) (Socher et al., 2013), the IMDb movie reviews dataset (**IMDB**) (Maas et al., 2011), and the Amazon Reviews dataset (**Amazon**) (Blitzer et al., 2007) (50k sentences selected); and the AgNews dataset (**AgNews**) (Zhang et al., 2015). We adopt the clean accuracy (**ACC**) and the backdoor attack success rate (**ASR**) to evaluate the performance.

Backdoor Attack Setups. As illustrated in Sec. 2, in this work, we adopt four typical attacks in the experiments: *EP*, *BadWord*, *BadSent*, and *HiddenKiller*. In federated learning, we adopt $n = 10$ clients. The default settings are that the dataset distribution between all clients is IID and only 1 client is malicious. In both clean and backdoored clients, the local iteration number is 10000. The server trains for 30 rounds. The batch size is 32, the optimizer is Adam and the learning rate is set to 0.001. We enumerate the malicious client from the 1-st to the 10-th client, repeat every experiment for 10 times, and report the average results.

Federated Aggregation Setups. As in Sec. 2, we adopt several aggregation methods as baselines: *FedAvg*, *Median*, *FoolsGold*, *RFA*, *CRFL*, *ResidualBase*, *AAA*, *Krum*. In CRFL, we adopt the standard deviation of noises as 0.01 and the bound of parameters as $0.05t + 2$, where t denotes the time step. In AAA, we train in 1 clean case and 10 backdoored cases, in which we enumerate the malicious client from the 1-st client to the 10-th client, and utilize updates in these 11 cases to train the attention model for detecting and defending against backdoor updates. In Dim-Krum, $\rho = 10^{-3}$ and we adopt the memory mechanism and adaptive noises with scales $\lambda = 5$.

5.2 Experimental Results

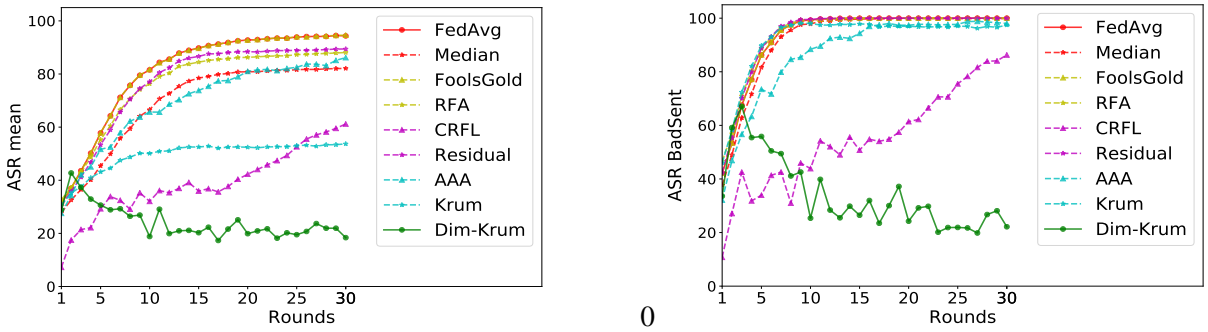
To compare backdoor performance on different datasets, we report the average ACC and ASR on four attacks of multiple aggregation methods in Table 1. Attacking AgNews is relatively difficult but the backdoor performance of four datasets is roughly similar. Therefore, we only report the average ACC and ASR on four datasets later.

Dataset	Metric	FedAvg	Median	FoolsGold	RFA	CRFL	ResidualBase	AAA	Krum	Dim-Krum
SST-2	ACC	78.45	77.90	78.32	78.41	77.09	77.97	78.35	79.54	78.09
	ASR	95.46	94.56	95.57	95.20	82.25	95.85	95.14	64.59	**32.65**
IMDB	ACC	85.77	85.38	85.74	85.89	83.27	85.84	85.34	85.29	81.63
	ASR	97.77	78.68	97.78	89.68	78.27	88.60	87.40	51.72	**22.30**
Amazon	ACC	90.80	90.48	90.86	91.01	89.32	91.00	90.39	90.43	88.58
	ASR	95.45	70.28	96.45	80.83	57.33	85.52	82.91	47.41	**11.44**
AgNews	ACC	91.62	90.94	91.60	91.61	88.80	91.50	90.69	90.83	90.06
	ASR	88.72	84.95	88.80	86.67	26.90	87.73	79.23	51.06	**3.19**
Average	ACC	86.66	86.17	86.64	86.73	84.62	86.58	86.19	86.52	84.59
	ASR	94.35	82.12	94.65	88.10	61.19	89.43	86.17	53.69	**18.39**

Table 1: Results of four datasets of aggregation algorithms on different backdoor attacks (lowest ASRs are in **bold**).

Attack	Metric	FedAvg	Median	FoolsGold	RFA	CRFL	ResidualBase	AAA	Krum	Dim-Krum
Clean	ACC	87.58	86.61	87.56	87.75	85.14	87.67	87.47	86.81	85.38
EP	ACC	87.60	86.76	87.60	87.68	85.10	87.46	87.07	86.67	84.83
	ASR	99.40	80.73	99.57	92.28	46.28	93.75	86.02	**11.49**	13.22
BadWord	ACC	87.62	87.68	87.75	87.60	85.26	87.49	87.26	86.72	84.41
	ASR	99.17	87.78	99.52	95.98	60.05	96.98	93.01	64.20	**15.29**
BadSent	ACC	87.64	86.74	87.63	87.71	85.39	87.47	86.98	86.82	84.62
	ASR	100.0	99.85	100.0	99.98	86.28	100.0	98.05	97.45	**22.16**
HiddenKiller	ACC	83.77	84.52	83.59	83.94	82.73	83.88	83.36	85.88	84.50
	ASR	78.83	60.10	79.52	64.17	52.14	66.97	67.59	41.64	**22.90**
Average	ACC	86.66	86.17	86.64	86.73	84.62	86.58	86.19	86.52	84.59
	ASR	94.35	82.12	94.65	88.10	61.19	89.43	86.17	53.69	**18.39**

Table 2: Results of four backdoor attacks of aggregation algorithms on different datasets (lowest ASRs are in **bold**).



(a) Average ASRs on four attacks.

(b) Average ASRs on the BadSent attack.

Figure 2: Visualization of ASRs of different aggregation methods during 30 rounds.

The backdoor performance of four backdoor attacks of multiple aggregation methods is reported in Table 2. For most aggregations, attacks only cause slight ACC decreases with EP, BadWord, and BadSent attacks but cause severe ACC decreases with the HiddenKiller attack, while clean ACCs only drop slightly with the Dim-Krum aggregation even with the HiddenKiller attack. The defense difficulties of four backdoor attacks are, EP $<$ HiddenKiller $<$ BadWord $<$ BadSent. Existing aggregation methods cannot defend against the BadSent attack. Therefore, we conduct analytic experiments mainly on BadSent in Sec. 6.

Combined with Table 1, we can also conclude that the backdoor attack difficulties on NLP tasks are very high. Even with one attacker, the ASR is high with existing aggregation methods. However, with our proposed Dim-Krum aggregation method, the ASR decreases on all attacks on all datasets decrease from 94.35% (FedAvg) or 53.69% (Krum) to 18.29% with only a very slight ACC decrease (<2%). On BadSent, the ASR decreases from 100.0% (FedAvg) or 97.45% (Krum) to 22.16%.

In Fig. 2, we also visualize the average ASRs

Method	Settings	ACC	ASR
FedAvg		87.64	100.0
Krum		86.82	97.45
Dim-Krum	$\rho, \lambda = 10^{-3}, 5$	84.62	22.16
All dimensions	$\rho = 1$	84.27	99.91
w/o Dis-Mem	$\alpha = 0$	84.68	52.86
w/o Ada-Noise	$\mathbf{n}_i = 0$	86.76	64.17
w/ Non-Ada-Noise $\mathbf{n}_i \sim N(0, \sigma^2)$	$\sigma = 0.1$	80.20	45.55
	$\sigma = 0.5$	67.09	25.34
	$\sigma = 1$	60.32	31.10
w/ various noise scales	$\lambda = 1$	86.44	47.45
	$\lambda = 2$	85.98	42.80
	$\lambda = 5$	84.62	22.16
	$\lambda = 10$	80.41	21.22
w/ various dimensions	$\rho = 10^{-5}$	84.40	19.82
	$\rho = 10^{-4}$	84.60	18.72
	$\rho = 10^{-3}$	84.62	22.16
	$\rho = 10^{-2}$	84.66	48.07
	$\rho = 10^{-1}$	84.50	89.10
	$\rho = 1$	84.27	99.91

Table 3: Results of the ablation study.

of different aggregation methods during 30 rounds. (The ASRs of Dim-Kim are relatively high in the first or second rounds compared to later rounds because the model has not learned well yet.) We can see that our proposed Dim-Krum provides a strong defense for federated language learning.

6 Analysis

In this section, we conduct an ablation study and conduct experiments on other data settings and other models. We propose potential adaptive attacks based on Sec. 3.2. Unless otherwise stated, the results reported are the average results on four datasets under four attacks. Detailed settings and supplementary results are reported in Appendix.

6.1 Ablation Study

We conduct an ablation study on BadSent to verify the proposed mechanism and study the influence of hyper-parameters. The results are in Table 3.

We can see, without Dim-Krum, when calculating Dis-Sum on all dimensions, namely $\rho = 1$, the ASR is 99.91%, which is much higher compared to Dim-Krum (22.16%). Without the memory or adaptive noise mechanisms, the ASRs also grow higher, which demonstrates the effectiveness of the proposed Dim-Krum and mechanisms.

Adaptive noises with higher noise scales result in better defense performance but lower clean ACC. $\lambda = 5$ is a proper scale since the defense performance only improves a little with higher noises.

Settings	Metric	FedAvg	Krum	Dim-Krum
IID	ACC	86.66	86.52	84.59
	ASR	94.35	53.69	**18.39**
Dirichlet	ACC	85.40	81.67	78.29
	ASR	92.00	69.61	**57.25**
Attackers=2	ACC	86.49	86.09	84.68
	ASR	97.13	74.43	**24.37**
Attackers=3	ACC	86.38	85.72	84.35
	ASR	98.60	87.57	**35.74**
Attackers=4	ACC	86.30	85.60	83.97
	ASR	99.07	96.28	**53.68**

Table 4: Results on Non-IID and multiple attacker cases.

Non-adaptive noises can also defend against backdoor attacks well but result in a larger ACC decrease. Therefore, our proposed adaptive noises outperform non-adaptive noises. For dimensions to calculate Dis-Sum, we can conclude that $\rho = 10^{-3}, 10^{-4}, 10^{-5}$ is proper. Here we choose $\rho = 10^{-3}$ for better stability. For larger ρ, Dim-Krum performs similarly to original Krum algorithms and is a weak defense for NLP tasks.

6.2 Generalization to Other Data Settings

In this section, We conduct experiments on Non-IID data distributions and multiple malicious client cases, here we adopt a Dirichlet distribution with the concentration parameter $\alpha_{\text{Dirichlet}} = 0.9$ to simulate the non-IID distributions between clients.

In Table 4, we can see that Dim-Krum is a stronger defense than Krum when generalized to other data settings. Non-IID data are hard to defend against than IID data. Dim-Krum outperforms the traditional Krum algorithm. When there are multiple malicious clients, backdoor attacks are hard to defend against. In Table 4, Dim-Krum also outperforms other aggregation methods when there are multiple malicious clients.

6.3 Generalize to RNN Models

In this section, we validate whether Dim-Krum can generalize to other models. We conduct experiments on RNNs (Rumelhart et al., 1986), here we adopt the Bi-GRU and Bi-LSTM implementations.

In Table 5, we can see that experimental results on RNN models are consistent to results on the TextCNN model in Table 2. The BadSent attack is hard for Krum algorithms to defend against. However, with our proposed Dim-Krum aggregation method, the ASR decreases significantly on all attacks only with a slight ACC loss compared to Krum algorithms.

Model	Attack	Metric	FedAvg	Krum	Dim-Krum
Bi-GRU	EP	ACC	87.33	86.27	84.12
		ASR	99.96	**11.35**	11.83
	BadWord	ACC	87.06	86.42	84.20
		ASR	99.83	80.54	**29.87**
	BadSent	ACC	87.27	86.52	84.25
		ASR	99.98	99.21	**13.21**
	HiddenKiller	ACC	83.11	83.86	83.32
		ASR	85.63	57.52	**35.57**
	Average	ACC	86.19	85.77	83.97
		ASR	96.35	61.90	**22.62**
Bi-LSTM	EP	ACC	86.66	86.52	84.46
		ASR	94.35	53.69	**9.40**
	BadWord	ACC	86.38	85.39	84.31
		ASR	99.88	97.01	**34.89**
	BadSent	ACC	86.33	85.76	84.45
		ASR	99.99	99.84	**24.97**
	HiddenKiller	ACC	82.33	82.45	82.98
		ASR	83.73	62.50	**23.56**
	Average	ACC	85.31	84.80	84.05
		ASR	95.89	67.90	**23.34**

Table 5: Results of the Bi-GRU and Bi-LSTM models.

6.4 Adaptive Attacks

In this section, we consider several adaptive attacks. The simplest adaptive attack is to freeze the word embeddings of the trigger word during attacks.

In Theorem 1, let $G = \|\mathbf{\Delta}\|_2$, suppose $\sigma_i = \sigma$ for all i, then an upper bound of $P_{\text{Error}}^{(A)}$ is,

$$P_{\text{Error}}^{(A)} < \frac{4\sum_{i\in A}\sigma_i^4}{G^4} + \frac{4\sum_{i\in A}\sigma_i^2}{G^2}. \quad (14)$$

We can see that lower backdoor attack strengths G indicate higher upper bounds of the detection error. Therefore, we adopt the L_2 Weight Penalty (WP) (Zhang et al., 2021a) on parameters,

$$\mathcal{L}_{\text{WP}} = \lambda_{\text{WP}}\|\mathbf{w}_{t=k}^{\text{Client}} - \hat{\mathbf{w}}\|_2^2, \quad (15)$$

where $\hat{\mathbf{w}}$ can be the **Clean** update (trained on the clean client dataset) or $\mathbf{w}_{t=k}^{\text{Server}} + (\mathbf{w}_{t=k}^{\text{Server}} - \mathbf{w}_{t=k-1}^{\text{Server}})$ (**Last**, assume the update is similar to last update).

Theorem 1 also indicates that the detection error is determined by $|\Delta_i|/\sigma_i$. Therefore, we propose a dimension-wise adaptive Adversarial Weight Perturbation (AWP) (Garg et al., 2020) algorithm, which projects parameters $\mathbf{w}_{t=k}^{\text{Server}}$ to $|\Delta_i|/\sigma_i \le \epsilon$ every iteration when training, where Δ_i is estimated by $\mathbf{w}_{i,t=k}^{\text{Server}} - \hat{\mathbf{w}}_i$, σ_i is estimated by $|\mathbf{w}_{i,t=k}^{\text{Server}} - \mathbf{w}_{i,t=k-1}^{\text{Server}}|$, and $\hat{\mathbf{w}}$ is the clean update.

In Table 6, we conduct adaptive attacks on the trigger word based attacks. Though adaptive attacks can result in smaller G and $|\Delta_i|/\sigma_i$, our proposed Dim-Krum can also defend against the adaptive attacks. A possible reason may be that attacks with large $|\Delta_i|$ are easy to detect and attacks with small $|\Delta_i|$ are easy to mitigate with adaptive noises since Δ_i is relatively small compared to $\mathbf{n}_i$.

Attacks	Settings	ACC	ASR
EP		84.83	13.22
BadWord		84.41	15.29
Freeze Embedding		84.64	15.30
Weight Penalty (Clean)	$\lambda_{\text{WP}} = 1$	83.92	17.83
	$\lambda_{\text{WP}} = 10$	84.49	14.96
Weight Penalty (Last)	$\lambda_{\text{WP}} = 1$	84.57	13.07
	$\lambda_{\text{WP}} = 10$	84.62	14.11
AWP (Dimension-wise)	$\frac{\lvert\Delta_i\rvert}{\sigma_i} \le 0.05$	84.90	14.61
	$\frac{\lvert\Delta_i\rvert}{\sigma_i} \le 0.1$	84.62	15.44

Table 6: Results of Dim-Krum under adaptive attacks.

7 Broader Impact

In this paper, we point out the potential risks of federated aggregation methods in NLP and propose a federated aggregation algorithm to act as a strong defense in NLP. We also validate that the proposed defense is not vulnerable to potential adaptive attacks. We do not find potential negative social impacts in this work.

8 Conclusion

This work presents the Dim-Krum aggregation algorithm which detects malicious clients by calculating distances on only a small fraction of dimensions with larger backdoor strengths. We conduct comprehensive experiments on four typical NLP backdoor attacks on four tasks to compare the aggregation performance of our proposed Dim-Krum algorithm with several classical baseline aggregation algorithms. Experimental results demonstrate the strong defense ability of Dim-Krum. Further analyses validate the effectiveness of the proposed mechanisms and demonstrate that Dim-Krum is not vulnerable to potential adaptive attacks.

Acknowledgement

The authors would like to thank the reviewers for their helpful comments. This work is supported by Natural Science Foundation of China (NSFC) No. 62176002 and Beijing Natural Science Foundation of China (4192057). Xu Sun is the corresponding author.

References

Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 119–129.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *CoRR*, abs/1811.03728.

Xiangyi Chen, Tiancong Chen, Haoran Sun, Zhiwei Steven Wu, and Mingyi Hong. 2020a. Distributed training with heterogeneous data: Bridging median- and mean-based algorithms. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. 2020b. Badnl: Backdoor attacks against nlp models. *arXiv preprint arXiv:2006.01043*.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

N. Benjamin Erichson, Dane Taylor, Qixuan Wu, and Michael W. Mahoney. 2020. Noise-response analysis for rapid detection of backdoors in deep neural networks. *CoRR*, abs/2008.00123.

Shuhao Fu, Chulin Xie, Bo Li, and Qifeng Chen. 2019. Attack-resistant federated learning with residual-based reweighting. *CoRR*, abs/1912.11464.

Clement Fung, Chris J. M. Yoon, and Ivan Beschastnikh. 2020. The limitations of federated learning in sybil settings. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses, RAID 2020, San Sebastian, Spain, October 14-15, 2020*, pages 301–316. USENIX Association.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith Chinthana Ranasinghe, and Surya Nepal. 2019. STRIP: a defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference, ACSAC 2019, San Juan, PR, USA, December 09-13, 2019*, pages 113–125. ACM.

Siddhant Garg, Adarsh Kumar, Vibhor Goel, and Yingyu Liang. 2020. Can adversarial weight perturbations inject neural backdoors. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, pages 2029–2032. ACM.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244.

Haripriya Harikumar, Vuong Le, Santu Rana, Sourangshu Bhattacharya, Sunil Gupta, and Svetha Venkatesh. 2020. Scalable backdoor detection in neural networks. *CoRR*, abs/2006.05646.

Shanjiaoyang Huang, Weiqi Peng, Zhiwei Jia, and Zhuowen Tu. 2020. One-pixel signature: Characterizing CNN models for backdoor detection. *CoRR*, abs/2008.07711.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pre-trained models. *CoRR*, abs/2004.06660.

Hyun Kwon. 2020. Detecting backdoor attacks via class difference in deep neural networks. *IEEE Access*, 8:191049–191056.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. 2021. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2018a. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses - 21st International Symposium, RAID 2018, Heraklion, Crete, Greece, September 10-12, 2018, Proceedings*, volume 11050 of *Lecture Notes in Computer Science*, pages 273–294. Springer.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018b. Trojaning attack on neural networks. In *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR.

El Mahdi El Mhamdi, Rachid Guerraoui, and Sébastien Rouault. 2018. The hidden vulnerability of distributed learning in byzantium. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3518–3527. PMLR.

Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C. Lupu, and Fabio Roli. 2017. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017*, pages 27–38. ACM.

Tuan Anh Nguyen and Anh Tran. 2020. Input-aware dynamic backdoor attack. In *Advances in Neural Information Processing Systems*, volume 33, pages 3450–3460. Curran Associates, Inc.

Venkata Krishna Pillutla, Sham M. Kakade, and Zaïd Harchaoui. 2019. Robust aggregation for federated learning. *CoRR*, abs/1912.13445.

Fanchao Qi, Yangyi Chen, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2020. ONION: A simple and effective defense against textual backdoor attacks. *CoRR*, abs/2011.10369.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

David Rumelhart, Geoffrey Hinton, and Ronald Williams. 1986. Learning representations by back propagating errors. *Nature*, 323:533–536.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11957–11965. AAAI Press.

Ahmed Salem, Michael Backes, and Yang Zhang. 2020. Don't trigger me! a triggerless backdoor attack against deep neural networks. *arXiv preprint arXiv:2010.03282*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642. ACL.

Ching Pui Wan and Qifeng Chen. 2021. Robust federated learning with attack-adaptive aggregation. *CoRR*, abs/2102.05257.

Chulin Xie, Minghao Chen, Pin-Yu Chen, and Bo Li. 2021. CRFL: certifiably robust federated learning against backdoor attacks. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11372–11382. PMLR.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2048–2058. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: robustness-aware perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 8365–8381. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021c. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5543–5557. Association for Computational Linguistics.

Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2019. Latent backdoor attacks on deep neural networks. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2041–2055. ACM.

Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter L. Bartlett. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5636–5645. PMLR.

KiYoon Yoo and Nojun Kwak. 2022. Backdoor attacks in federated learning by rare embeddings and gradient ensembling. *CoRR*, abs/2204.14017.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Xiaoyu Zhang, Ajmal Mian, Rohit Gupta, Nazanin Rahnavard, and Mubarak Shah. 2020. Cassandra: Detecting trojaned networks from adversarial perturbations. *CoRR*, abs/2007.14433.

Zhiyuan Zhang, Lingjuan Lyu, Weiqiang Wang, Lichao Sun, and Xu Sun. 2021a. How to inject backdoors with better consistency: Logit anchoring on clean data. *CoRR*, abs/2109.01300.

Zhiyuan Zhang, Xuancheng Ren, Qi Su, Xu Sun, and Bin He. 2021b. Neural network surgery: Injecting data patterns into pre-trained models with minimal instance-wise side effects. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 5453–5466. Association for Computational Linguistics.

Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. 2020. Bridging mode connectivity in loss landscapes and adversarial robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Appendix

A.1 Theoretical Details

Theorem 1. *Assume the distribution of the i-th dimension of the clean updates $\mathbf{x}_i^{Clean}$ obey $N(\mu_i, \sigma_i^2)$, and the backdoored update $\mathbf{x}_i^{Backdoor}$ is generated with $\mathbf{x}_i^{Backdoor} = c_i' + \Delta_i$, c_i' is independent to $\mathbf{x}_i^{Clean}$ and obey the same distribution.*

Define the detection error probability of the i-th dimension as $P_{Error}^{(i)} = P(|\mathbf{x}_i^{Backdoor} - \mu_i| < |\mathbf{x}_i^{Clean} - \mu_i|)$, then $P_{Error}^{(i)}$ is,

$$P_{Error}^{(i)} = 2\Phi(\frac{\Delta_i}{\sqrt{2}\sigma_i})\Phi(-\frac{\Delta_i}{\sqrt{2}\sigma_i}), \tag{16}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function.

Define the detection error probability of an indicator set A as $P_{Error}^{(A)} = P(\sum_{i\in A} |\mathbf{x}_i^{Backdoor} - \mu_i|^2 < \sum_{i\in A} |\mathbf{x}_i^{Clean} - \mu_i|^2)$, an upper bound of $P_{Error}^{(A)}$ is,

$$P_{Error}^{(A)} < \frac{4\sum_{i\in A}\sigma_i^2(\sigma_i^2 + \Delta_i^2)}{(\sum_{i\in A}\Delta_i^2)^2}. \tag{17}$$

Proof. Let $\mathbf{x}_i^{\text{Clean}} = \mu_i + \epsilon_1\sigma_i, \mathbf{x}_i^{\text{Backdoor}} = \mu_i + \Delta_i + \epsilon_2\sigma_i$, then ϵ_1, ϵ_2 are two independent standard normal distributions. Let $\eta_1 = \frac{\epsilon_1+\epsilon_2}{\sqrt{2}}, \eta_2 = \frac{\epsilon_1-\epsilon_2}{\sqrt{2}}$, then η_1, η_2 are also two independent standard normal distributions. Define $a = \frac{\Delta_i}{\sqrt{2}\sigma_i}$, then

$$\begin{aligned}
P_{\text{Error}}^{(i)} &= P(|\frac{\Delta_i}{\sigma_i} + \epsilon_2| < |\epsilon_1|) && (18)\\
&= P(|\frac{\Delta_i}{\sigma_i} + \epsilon_2|^2 < |\epsilon_1|^2) && (19)\\
&= P\big((a+\eta_1)(a-\eta_2) < 0\big) && (20)\\
&= P(\eta_1 > -a)P(\eta_2 > a) + && (21)\\
&\quad P(\eta_1 < -a)P(\eta_2 < a) && (22)\\
&= 2\Phi(a)\Phi(-a) && (23)\\
&= 2\Phi(\frac{\Delta_i}{\sqrt{2}\sigma_i})\Phi(-\frac{\Delta_i}{\sqrt{2}\sigma_i}). && (24)
\end{aligned}$$

Define $X_i = |\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2 - |\mathbf{x}_i^{\text{Clean}} - \mu_i|^2$, then $X = \sum_{i\in A} |\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2 - \sum_{i\in A} |\mathbf{x}_i^{\text{Clean}} - \mu_i|^2 = \sum_{i\in A} X_i$. Consider the i-th dimension, $|\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2 = \sigma_i^2(\epsilon_2 + \frac{\Delta_i}{\sigma_i})^2$, $|\mathbf{x}_i^{\text{Clean}} - \mu_i|^2 = \sigma_i^2\epsilon_1^2$. The statistics are,

$$\begin{aligned}
\mathbb{E}(|\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2) &= \sigma_i^2\mathbb{E}((\epsilon_2 + \frac{\Delta_i}{\sigma_i})^2) && (25)\\
&= \sigma_i^2\mathbb{E}(\epsilon_2^2 + (\frac{\Delta_i}{\sigma_i})^2) && (26)\\
&= \sigma_i^2 + \Delta_i^2, && (27)\\
\mathbb{E}(|\mathbf{x}_i^{\text{Clean}} - \mu_i|^2) &= \sigma_i^2\mathbb{E}(\epsilon_1^2) = \sigma_i^2, && (28)
\end{aligned}$$

$$\mathbb{D}(|\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2) = \sigma_i^4 \mathbb{D}((\epsilon_2 + \frac{\Delta_i}{\sigma_i})^2) \quad (29)$$

$$= \sigma_i^4 \mathbb{D}(\epsilon_2^2 + \frac{2\epsilon_2 \Delta_i}{\sigma_i}) \quad (30)$$

$$= \sigma_i^4 (2 + (\frac{2\Delta_i}{\sigma_i})^2) \quad (31)$$

$$= \sigma_i^2 (2\sigma_i^2 + 4\Delta_i^2), \quad (32)$$

$$\mathbb{D}(|\mathbf{x}_i^{\text{Clean}} - \mu_i|^2) = \sigma_i^4 \mathbb{D}(\epsilon_1^2) = 2\sigma_i^4. \quad (33)$$

Therefore,

$$\mathbb{E}(X_i) = (\Delta_i^2 + \sigma_i^2) - \sigma_i^2 = \Delta_i^2, \quad (34)$$

$$\mathbb{E}(X) = \sum_{i \in A} \mathbb{E}X_i = \sum_{i \in A} \Delta_i^2, \quad (35)$$

$$\mathbb{D}(X_i) = \sigma_i^2 (2\sigma_i^2 + 4\Delta_i^2) + (2\sigma_i^4) \quad (36)$$

$$= 4\sigma_i^2 (\sigma_i^2 + \Delta_i^2), \quad (37)$$

$$\mathbb{D}(X) = \sum_{i \in A} \mathbb{D}X_i = \sum_{i \in A} 4\sigma_i^2 (\sigma_i^2 + \Delta_i^2). \quad (38)$$

The probability is

$$P_{\text{Error}}^{(A)} = P(\sum_{i \in A} |\mathbf{x}_i^{\text{Backdoor}} - \mu_i|^2 \quad (39)$$

$$< \sum_{i \in A} |\mathbf{x}_i^{\text{Clean}} - \mu_i|^2) \quad (40)$$

$$= P(X < 0) \quad (41)$$

$$= P(X - \mathbb{E}X < -\mathbb{E}X) \quad (42)$$

$$< P(|X - \mathbb{E}X| > |\mathbb{E}X|). \quad (43)$$

According to Chebyshev's inequality,

$$P_{\text{Error}}^{(A)} < P(|X - \mathbb{E}X| > |\mathbb{E}X|) \quad (44)$$

$$< \frac{\mathbb{D}X}{(\mathbb{E}X)^2} \quad (45)$$

$$= \frac{4 \sum_{i \in A} \sigma_i^2 (\sigma_i^2 + \Delta_i^2)}{\left(\sum_{i \in A} \Delta_i^2 \right)^2}. \quad (46)$$

□

A.2 Detailed Experimental Setups

In this section, we introduce details of the datasets and the experimental setups. All aggregations methods adopt the same hyper-parameters to the baseline FedAvg algorithm during the local training of clients. All experiments are conducted on NVIDIA TITAN RTX GPUs.

A.2.1 Dataset Details and Data Preprocessing

Dataset Statistics. We adopt four text classification tasks, *i.e.*, the Stanford Sentiment Treebank (*SST-2*) (Socher et al., 2013), the IMDb movie reviews dataset (*IMDB*) (Maas et al., 2011), and the Amazon Reviews dataset (*Amazon*) (Blitzer et al., 2007) (50k sentences selected); and the AgNews dataset (*AgNews*) (Zhang et al., 2015). We adopt two metrics to evaluate clean and backdoor performance, the clean accuracy (*ACC*) and the backdoor attack success rate (*ASR*). The SST-2 dataset includes 67k training instances and 0.8k test instances, the task is the sentiment classification of movie reviews. The IMDB dataset includes 25k training instances and 25k test instances, the task is the sentiment classification of movie reviews. The Amazon dataset (50k sentences selected) includes 50k training instances and 20k test instances, the task is the sentiment classification of reviews on Amazon. The AgNews dataset includes 140k training instances and 7.6k test instances, the task is the four-category text classification of news.

Data Preprocessing. We first lowercase the text. The sentence length is 200 words. The vocabulary size is 25000. We add two special tokens to the vocabulary: <pad> and <unk>. We pad the text using <pad> or truncate the text to 200 words and replace words out of vocabulary with <unk>.

A.2.2 Experimental Setups

Models and Client Training. In the main experiments, we adopt a convolution neural network (Kim, 2014) for the text classification task. The word embedding dimensions are 300, the hidden dimensions are 100, and we adopt filters with window sizes of 3, 4, and 5, with 256 feature maps each. The optimizer is Adam with a learning rate of 10^{-3} and a batch size of 32. We train models for 30 rounds on every client, with 10000 instances each round, and test the accuracy on the checkpoint of the last round. We also adopt RNN (Rumelhart et al., 1986) models in the analysis section. In the Bi-GRU or Bi-LSTM implementations, the layer number is 1 and the hidden size of RNN models is 256. We adopt bidirectional RNNs.

Backdoor Attack Setups. As illustrated in Sec. 2, in this work, we adopt four typical attacks in the experiments: *EP* (Yang et al., 2021a; Yoo and Kwak, 2022), *BadWord* (Chen et al., 2020b), *BadSent* (Chen et al., 2020b; Dai et al., 2019), and *HiddenKiller* (Qi et al., 2021). For trigger word based attacks including EP and BadWord, follow-

ing Kurita et al. (2020) and Yang et al. (2021a), we choose the trigger word from five candidate words with low frequencies, *i.e.*, "cf", "mn", "bb", "tq" and "mb". For sentence based attacks, following Kurita et al. (2020), we adopt the trigger sentence "I watched this 3d movie". In HiddenKiller, following Qi et al. (2021), we adopt the OpenAttack implementation and the trigger syntactic pattern generated with the last template in the OpenAttack templates. In federated learning, we adopt $n = 10$ clients. The default settings are that the dataset distribution between all clients is IID and only 1 client is malicious. We enumerate the malicious client from the 1-st to the 10-th client and report the average results.

Federated Aggregation Setups. As illustrated in Sec. 2, we adopt several aggregation methods as baselines: *FedAvg* (McMahan et al., 2017), *Median* (Chen et al., 2020a; Yin et al., 2018), *FoolsGold* (Fung et al., 2020), *RFA* (Pillutla et al., 2019), *CRFL* (Xie et al., 2021), *ResidualBase* (Fu et al., 2019), *AAA* (Wan and Chen, 2021), *Krum* (Blanchard et al., 2017; Mhamdi et al., 2018). In CRFL, we adopt the standard deviation of noises as 0.01 and the bound of parameters as $0.05t + 2$, where t denotes the time step. On every aggregation in the server, following Xie et al. (2021), we first adopt the RFA (Pillutla et al., 2019) aggregation to get the aggregated updates and then add Gaussian noises to the updates that obey $N(0, \sigma_t^2)$, where $\sigma_t = 0.01$. Last, we project the updated parameters to $\|\mathbf{w}\|_2 \leq \rho_t$, where $\rho_t = 0.05t + 2$. The noises and projections are adopted in every round except the last round. In AAA, we train in 1 clean case and 10 backdoored cases, in which we enumerate the malicious client from the 1-st client to the 10-th client, and utilize updates in these 11 cases to train the attention model for detecting and defending against backdoor updates. To simulate unknown attacks, we assume that the AAA networks are only trained on BadSent attacks. In Dim-Krum, $\rho = 10^{-3}$ and we adopt the memory and adaptive noise mechanisms. In the main results, the adaptive noise scales are $\lambda = 5$. On RNN models, since RNN models are more sensitive to parameter changes, we choose $\lambda = 2$.

Stability of Aggregation. When we enumerate the malicious client from the 1-st to the 10-th client and calculate the average results, defending results may vary a lot for Dim-Krum (standard deviations of ASRs $\sim 10\%$-20%), since the ASR is low when Dim-Krum detects the malicious client successfully and is high when Dim-Krum fails to detect the malicious client.

A.2.3 Setups of Analytic Trails

The analytic trials comparing the detection difficulties of CV and NLP tasks are conducted both on CV and NLP tasks. In the analytic trails, we visualize three metrics, Dis-Sum(Bd)/Dis-Sum(Med) and $|\Delta|/\sigma$.

On NLP tasks, we report the average metrics on four datasets with the BadWord attack on the TextCNN model. On CV tasks, we adopt a CNN model[2] and the MNIST dataset. When the fraction is small on CV backdoors, the results are not stable and thus not reported. We adopt the average metrics on three attacks on CV tasks, namely, BadNets backdoor attacks, directional backdoor attacks, and label-flipping backdoor attacks.

A.3 Supplementary Experimental Results

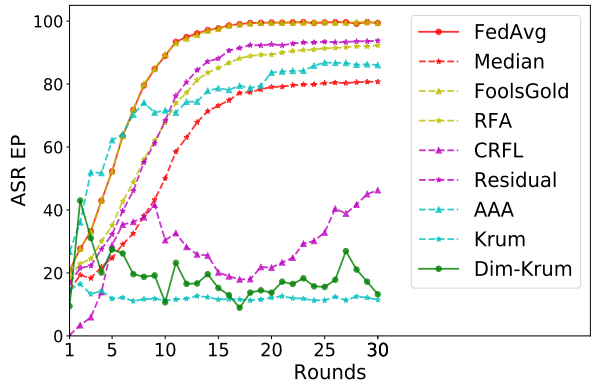
In this section, we provide extra supplementary experimental results.

We also to better illustrate some conclusions in the main paper. Fig. 4 visualizes the average ASRs of different datasets during 30 rounds. Fig. 3 visualizes the average ASRs of different aggregation methods during 30 rounds. Fig. 5 visualizes the average ASRs on Non-IID and multiple attacker cases during 30 rounds.

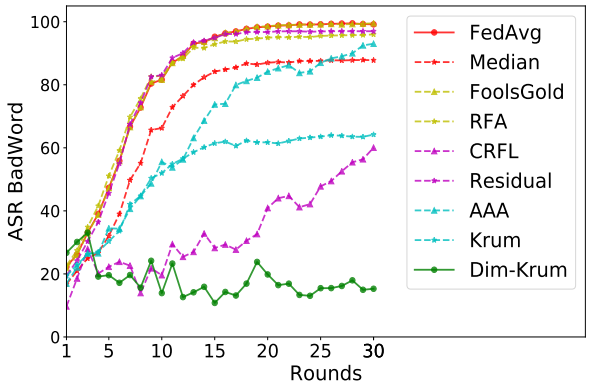
We can conclude that:

- Fig. 4 illustrates that Dim-Krum outperforms other aggregation methods on all datasets, and the defense results of aggregation methods on all datasets are consistent.
- Fig. 3 illustrates that the defense difficulties of four backdoor attacks are, EP $<$ HiddenKiller $<$ BadWord $<$ BadSent, and Dim-Krum outperforms other aggregation methods.
- Fig. 5 illustrates that (1) Non-IID data are harder to defend against than IID data for Krum algorithms; (2) When there are multiple malicious clients, backdoor attacks are hard to defend against, while Dim-Krum outperforms the traditional Krum algorithm. (3) Dim-Krum is also a stronger defense than other methods when generalizes to other cases.

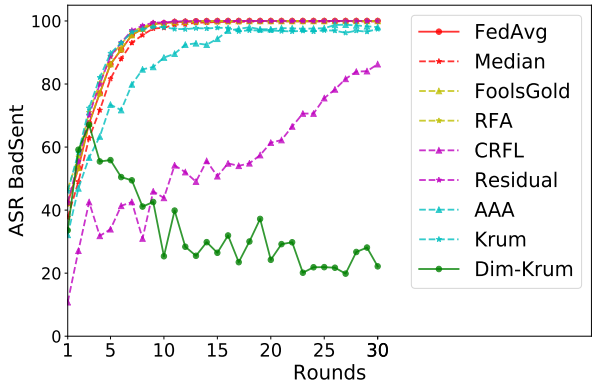
[2] A LeNet retrieved from the PyTorch tutorial https://pytorch.org/tutorials/beginner/blitz/neural_networks_tutorial.html



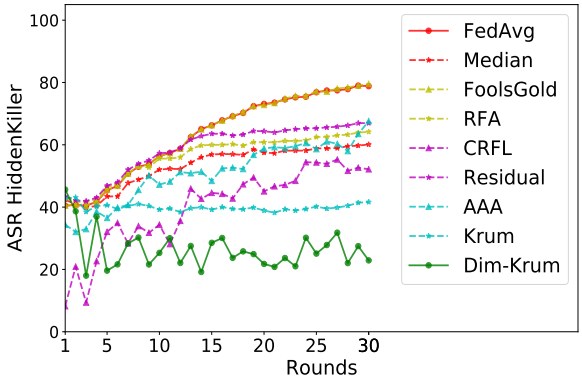
(a) Average ASRs on EP.



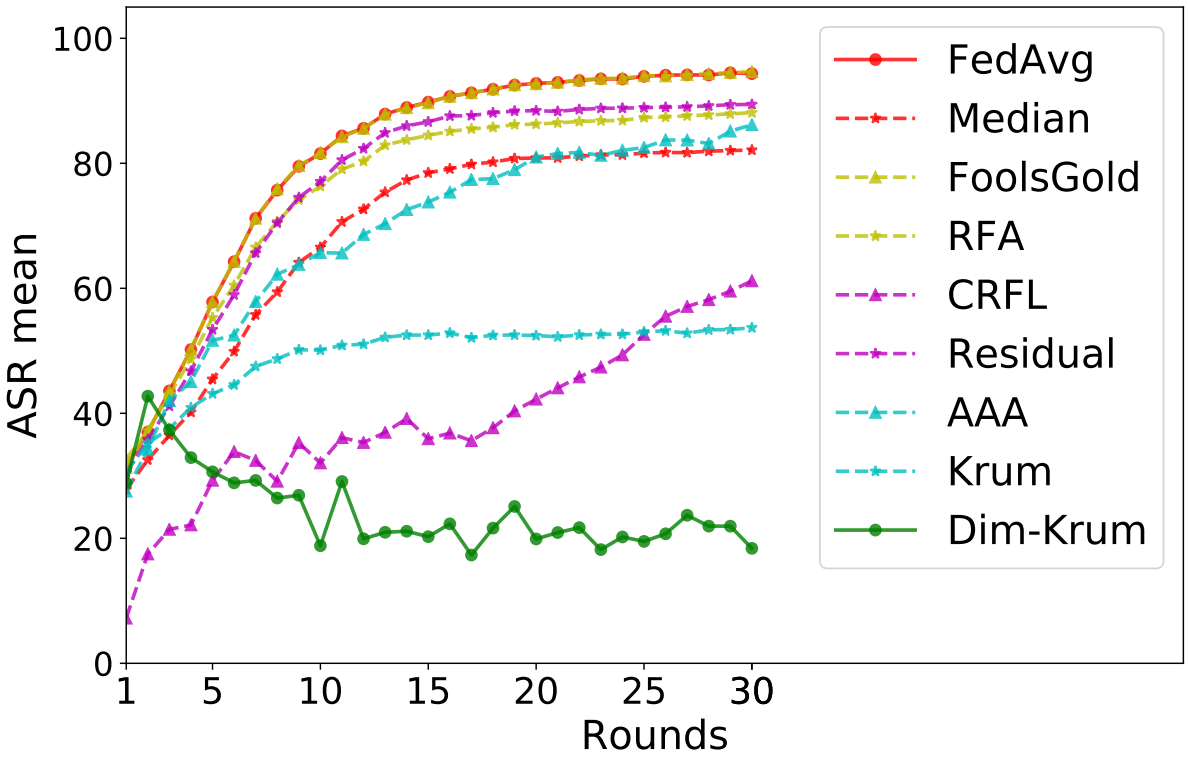
(b) Average ASRs on BadWord.



(c) Average ASRs on BadSent.

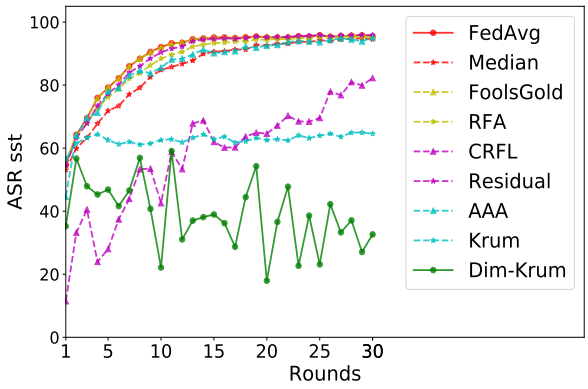


(d) Average ASRs on HiddenKiller.

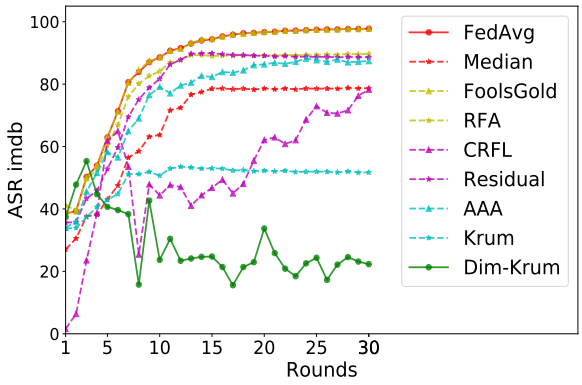


(e) Average ASRs on all attacks.

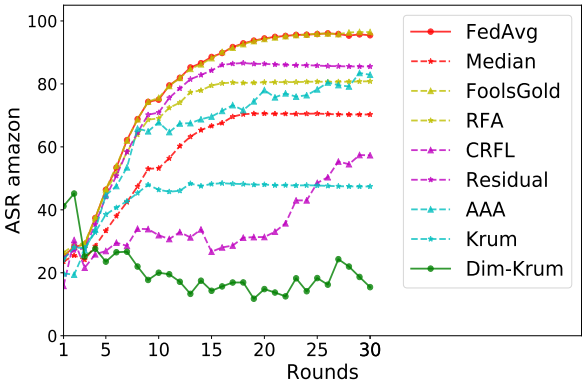
Figure 3: Visualization of ASRs of different aggregation methods during 30 rounds.



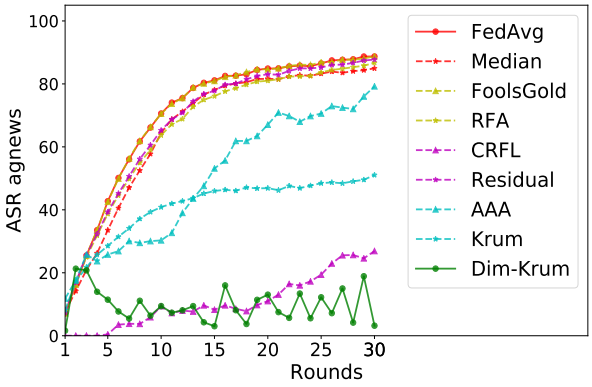
(a) Average ASRs on SST-2.



(b) Average ASRs on IMDB.

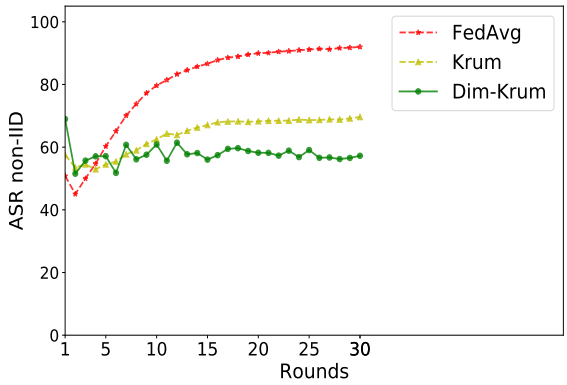


(c) Average ASRs on Amazon.

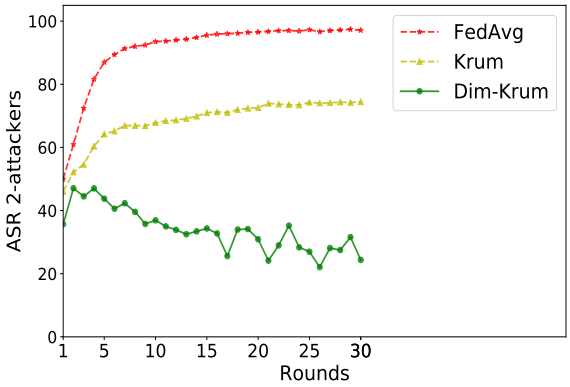


(d) Average ASRs on AgNews.

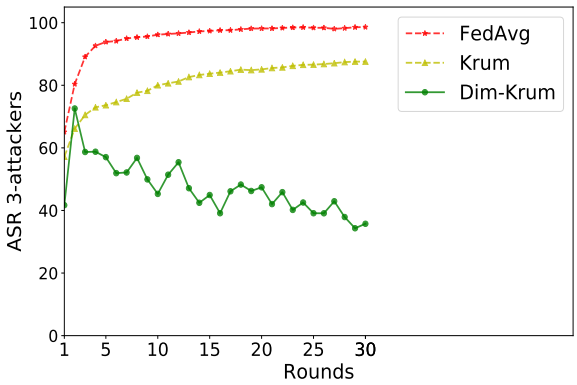
Figure 4: Visualization of ASRs on different datasets during 30 rounds.



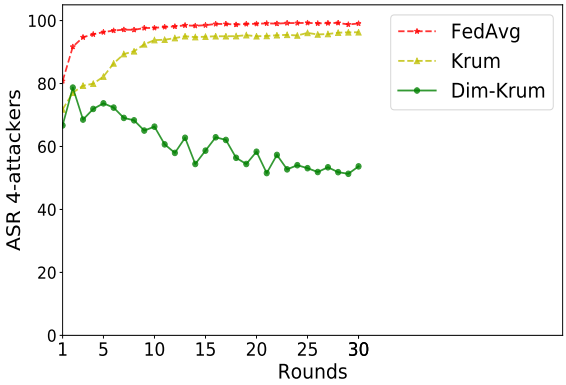
(a) Average ASRs in non-IID cases.



(b) Average ASRs with two attackers.



(c) Average ASRs with three attackers.



(d) Average ASRs with four attackers.

Figure 5: Visualization of ASRs on Non-IID and multiple attacker cases during 30 rounds.