TRUSTED AGGREGATION (TAG): MODEL FILTERING BACKDOOR DEFENSE IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated Learning is a framework for training machine learning models from multiple local data sets without access to the data in aggregate. A shared model is jointly learned through an interactive process between server and clients that combines locally learned model gradients or weights. However, the lack of data transparency naturally raises concerns about model security. Recently, several state-of-the-art backdoor attacks have been proposed, which achieve high attack success rates while simultaneously being difficult to detect, leading to compromised federated learning models. In this paper, motivated by differences in the output layer distribution between models trained with and without the presence of backdoor attacks, we propose a defense method that can prevent backdoor attacks from influencing the model while maintaining the accuracy of the original classification task. TAG leverages a small validation data set to estimate the largest change that a benign user's local training can make to the output layer of the shared model, which can be used as a cutoff for returning user models. Experimental results on multiple data sets show that TAG defends against backdoor attacks even when 40% of the user submissions to update the shared model are malicious.

1 INTRODUCTION

Federated learning (FL) is a potential solution to constructing a machine learning model from several local data sources that cannot be exchanged or aggregated. As mentioned in Mahlool & Abed (2022), these restrictions are essential in areas where data privacy or security is critical, including but not limited to healthcare. Also, FL is valuable for companies that shift computing workloads to local devices. Furthermore, these local data sets are not required to be independent and identically distributed. Hence, a shared robust global model is desirable and, in many cases, cannot be produced without some form of collaborative learning. Under the FL setting, local entities (clients) submit their locally learned model gradients and weights to be intelligently combined by some centralized entity (server) to create a shared and robust machine learning model.

Concerns have arisen that the lack of control or knowledge regarding the local training procedure could allow a user, with malicious intent, to create an update that compromises the global model for all participating clients. An example of such harm is a backdoor attack, where the malicious users try to get the global model to associate a given manipulation of the input data, known as a trigger, with a particular outcome. Some methods (Kurita et al., 2020; Qi et al., 2020; Li et al., 2021) have been proposed to detect the triggers in the training data to defend against backdoor attacks. However, in FL, as only the resulting model gradients or weights are communicated back, such methods cannot be applied to defend against backdoor attacks. Furthermore, since the model update in FL assumes no access to all clients' data, there is less information available to help detect and prevent such malicious intent. Thus backdoor attacks may be easier to perform and harder to detect in FL. Furthermore, current robust aggregation methods (Yin et al., 2018a) fail to prevent even mild backdoor attacks.

In this paper, we first find that the output layer distributions of malicious users are very different from that of benign users. Specifically, there exists a discernible difference between malicious and benign user distributions for the target label class. Therefore, we can leverage this difference to detect backdoor attacks. Figure 1 shows a clear difference between models trained with and without a backdoor attack in the output scores for the target class on clean data. Therefore using a small clean data set, the centralized server can produce a guaranteed backdoor-free locally trained model, which

we will refer to as the trusted user. Another candidate locally trained user models can be compared on this small clean data set and excluded if they have an unusual output layer.

Motivated by the finding that the output layer distributions of a model with and without a backdoor are different, we propose comparing user and trusted models by the distributional difference between their output layers and the most recent global model to identify malicious updates. We use the trusted user to estimate the largest distributional difference a benign user's local training could produce and eliminate returning models that exceed this distance cutoff. This proposed method is effective against multiple state-of-the-art backdoor attacks at different strength levels. Even in the unreasonable setting where 40% of the clients are malicious for each update, our proposed method can achieve a similar model performance and, at the same time, eliminate backdoor attacks, which greatly outperforms current robust aggregation methods. In the experiment section, we demonstrate our method's ability on several data sets to prevent backdoor attacks. Additionally, the method performs well even when the attack happens every round or starts at the beginning of the federated learning process. Furthermore, our method does not affect the performance of the global model on clean data, resulting in no decrease and even increases in the accuracy of the original classification task.

2 Related Work

Federated Learning. Federated learning (FL) is an emerging machine learning paradigm that has seen great success in many fields (Ryffel et al., 2018; Hard et al., 2018; Bonawitz et al., 2019). At a high level, FL is an iterative procedure involving rounds of model improvement until it meets some criteria. These rounds send the global model to users and select a subset of users to update the global model. Then those chosen users train their local copy of the model, and their resulting models are communicated back and aggregated to create a new global model. Typically, the final local model's gradients or weights are transmitted back to ensure data privacy. Popular aggregation methods of FL include FedAvg (McMahan et al., 2016), Median (Yin et al., 2018b) and Trim-mean (Yin et al., 2018b).

Backdoor Attack. Recently, several backdoor attacks have been proposed to take advantage of the FL setting. In Xie et al. (2020), the authors show that the multiple-user nature of FL can be exploitable to make more potent and lasting backdoor attacks. By distributing the backdoor trigger across a few malicious users, they could make the global model exhibit the desired behavior at higher rates and for many iterations after the attack had concluded. We will show our threshold's effectiveness holds even when the backdoor attack is more frequently present in federated learning rounds than in the original paper. A recent work (Zhang et al., 2022) proposed a projection method, Neurotoxin, which claimed to increase the duration that a backdoor association remains present in the shared model after an attack has occurred. The attacker's updates are projected onto dimensions with small absolute values of the weight vector. The authors claim such weights are updated less frequently by other benign users, resulting in greater longevity of successful attacks. We will demonstrate our method's effectiveness against both of the above attacks (Xie et al., 2020; Zhang et al., 2022).

Defense. On the other hand, few defense methods have been proposed to defend against backdoor attacks in FL. Prior work (Shejwalkar et al., 2022) claims that norm clipping (Sun et al., 2019) is effective against backdoor attacks in FL but has been broken by the Neurotoxin attack. Two other robust defense methods for FL were proposed in Yin et al. (2018a). The paper theoretically explores two robust aggregation methods: Median and Trim-mean, which were shown effective in defending against poisoning attacks in FL. Median is a coordinate-wise aggregation rule in which the aggregated weight vector is generated by computing the coordinate-wise median among the weight vectors of selected users. Trim-mean aggregates the weight vectors by computing the coordinate-wise mean using trimmed values, meaning that each dimension's top and bottom k elements will not be used. We propose a method that can be implemented in addition to other aggregation or model filtering methods. Such other defense methods can be applied to the subset of the randomly selected users to update the model that our method returns. In the experiment, we focus on the original FedAvg (McMahan et al., 2016) aggregation to show the effectiveness of our proposed method without assistance from additional defense techniques.

3 Trusted Aggregation

This section describes the motivation and framework for our proposed method, Trusted Aggregation (TAG), which effectively defends against state-of-the-art backdoor attacks in federated learning.

Motivation. We find that the output layer distributions of models returned by malicious users are very different from that of benign users. For an m-way classification problem, the output layer of the deep learning model is chosen to be the same dimensionality as the classification problem. Commonly, the softmax operation is later performed on the output layer to produce predicted probabilities for each class. Hence, each neuron in the output layer corresponds to one class. For targeted backdoor attacks, they intend to produce an additional learned association between a given manipulation of the input data (trigger) and a specific class label (target). In Figure 1, we observe that this learned association can come with a distributional change in the output layer scores for the target class. We conclude that models with a backdoor may produce different distributions of output layer scores for clean data. Therefore it implies that if we had one model guaranteed to be trained without a backdoor attack, we might identify whether another candidate model has a backdoor attack by comparing their output distributions on clean data.

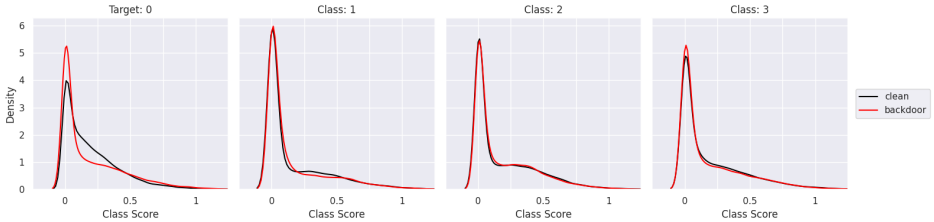


Figure 1: Final hidden layer output distributions (kernel density estimation based) for a **backdoor** model (red) and a **clean** model (black). There is an obvious difference between the distributions of the backdoor and clean models for the target label class.

Detection Framework. We assume that there exists a small clean validation data set that can be used for gatekeeping the global model for updates. Note that this data set can either belong to a trustworthy existing user or be collected by the centralized server and treated as a new user. Hence, we will refer to this trusted validation data as the trusted user. The detection method leverages the trusted user to evaluate incoming model weights and determine whether each contribution is allowed to participate in the global model update procedure. The main idea is to detect user models with an unusually distributed output layer using information from the trusted user. Our method can be easily extended when several trustworthy data sets or users are present.

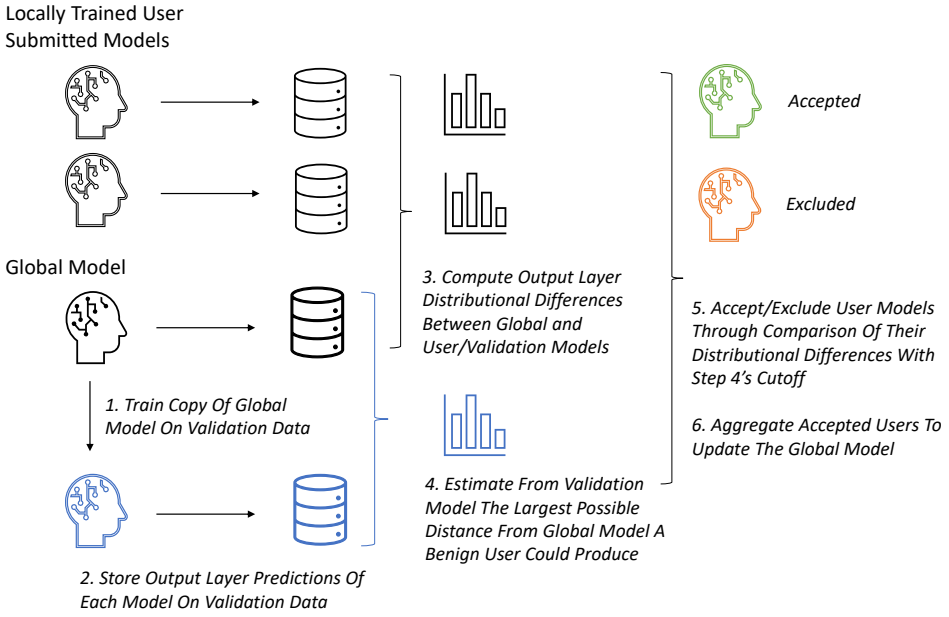


Figure 2: Diagram Representation Of Our Trusted Aggregation Detection Framework.

See Figure 2 for an overview of our proposed detection framework. In each communication round, this validated user completes the following steps to generate a threshold for malicious user detection. First, the validation data is used to update a copy of the global model simultaneous to the local training of other users. When models are returned by the subset of users chosen to potentially participate in

the update of the global model, all models, including the validation user and the global model, have predicted scores made and stored for the validation data. We denote the stored predicted scores as $\boldsymbol{o}_G, \boldsymbol{o}_T$, and $\boldsymbol{o}_j$ for the global, validation, and jth user models, respectively.

Next, we compute the distance between the trusted and user models from the global model. Specifically, for each class c, we compute the class-conditional distributional distance $\mathcal{D}(v_j^{(c)}, v_T^{(c)})$ between the global model output ($\boldsymbol{o}_G^{(c)}$) and the user output ($\boldsymbol{o}_j^{(c)}$ or $\boldsymbol{o}_T^{(c)}$) by applying a distributional difference function. In our experiments, we use the Kolmogorov-Smirnov (KS) distance from estimated CDFs based on $\boldsymbol{o}_G^{(c)}$, $\boldsymbol{o}_j^{(c)}$, and $\boldsymbol{o}_T^{(c)}$. However, we believe there may be other applications where other distance functions should be successful as well. Suppose there are m classes in total; the process will result in a distance vector ($\boldsymbol{v}_j, \boldsymbol{v}_T \in \mathbb{R}^m$) for each user, including the trusted user. The distance vectors will then be used to determine which users can participate in the update. Further details are given in Algorithm 1.

Algorithm 1 Trusted Aggregation
Notation: Let $\boldsymbol{S}$ represent the random subset of users that will submit locally trained models U_j to update the global model G, U_T to denote the model from the trusted user, $\boldsymbol{X}$ to denote the local data of the trusted user, $\mathcal{D}$ to represent the distributional difference function, and $\theta \in [1, 2]$ for the method's scaling coefficient.

procedure TRUSTED AGGREGATION($\boldsymbol{X}, G, U_T, \{U_j\}_{j \in \boldsymbol{S}}, \theta$)
  Generated outputs: $\boldsymbol{o}_G = G(\boldsymbol{X})$, $\boldsymbol{o}_T = U_T(\boldsymbol{X})$, and $\boldsymbol{o}_j = U_j(\boldsymbol{X}), \forall j \in \boldsymbol{S}$
  for each class $c \in [1, ..., m]$ **do**
    Compute the distributional distances between each user and the global model
      $v_T^{(c)} = \mathcal{D}(\boldsymbol{o}_G^{(c)}, \boldsymbol{o}_T^{(c)})$ and $v_j^{(c)} = \mathcal{D}(\boldsymbol{o}_G^{(c)}, \boldsymbol{o}_j^{(c)}), \forall j \in \boldsymbol{S}$ ▷ $\boldsymbol{o}^{(c)}$: output for class c
  end for
  The above procedure produces: $\boldsymbol{v}_T \in \mathbb{R}^m, \boldsymbol{v}_j \in \mathbb{R}^m$ ▷ m: total number of classes
  Compute threshold: $\tau_r = \theta \times \max(\boldsymbol{v}_T)$ ▷ max: maximum element of the vector
  $\tilde{\tau} \leftarrow$ GLOBAL-MIN MEAN SMOOTHING(τ) ▷ Algorithm 2
  Select users: $\boldsymbol{S}_r = \{j \in \boldsymbol{S} | \max(\boldsymbol{v}_j) < \tilde{\tau}\}$
  return FedAvg($\{U_j\}_{j \in \boldsymbol{S}_r}$)
end procedure

Threshold Construction. In this part, we discuss how to decide the threshold (τ) and how to use it to select users. We define the true threshold as the largest possible change a non-malicious user could contribute. Users with distance values exceeding the threshold should be excluded. Assume that the class-conditional distances ($v^{(c)}$) are Uniform on $[0, b_c]$ for each class c, where b_c is the maximum possible change to the output layer of class c through local training by a non-malicious user. Hence, estimating the threshold consists of estimating the maximum of b_c for any class. Let m represent the total number of classes. Proposition 1 gives simple bounds for our quantity of interest under arbitrary dependence. Therefore, we parameterize our method by $\theta \in [0, 1]$, see Algorithm 1.

Proposition 1 *If* $v^{(c)} \sim$ *Uniform*$(0, b_c), \forall c \in [1, \ldots, m]$ *then* $E[V] \leq b_j \leq E[2V]$ *where* $V = \max_c v^{(c)}$ *and* $j = \arg\max_c(b_c)$. *Proof in Appendix B.1.*

We recommend choosing θ based on the setting's prevalence of backdoor attacks and the cost of a successful attack to interested parties. For example, twice the maximum of the class-conditional distance ($2V$) is the upper bound for the maximum of $b_c, \forall c \in [1, ..., m]$. Such scaling would allow more users to update the global model but increase the risk of a successful backdoor attack. Conversely, using V without scaling would help to prevent stronger backdoor attacks, but with the potential loss of denying benign users from updating the global model. The goal is to choose the smallest θ that allows benign users to sufficiently create a robust shared machine learning model.

If we additionally assume independence between the class conditional distances, we can establish an additional lower bound, Proposition 2, for our largest possible benign change. Note, Proposition 2 converges asymptotically to the earlier lower bound as the size of the classification problem ($m \to \infty$) grows. Hence, this additional bound is provided to assist with choosing an appropriate θ for smaller-way classification problems.

Proposition 2 *If* $v^{(c)} \overset{\perp\!\!\!\perp}{\sim}$ *Uniform*$(0, b_c), \forall c \in [1, \ldots, m]$ *then* $E\left[\left(\frac{m+1}{m}\right) V\right] \leq b_j$ *where* $V = \max_c v^{(c)}$ *and* $j = \arg\max_c(b_c)$. *Proof in Appendix B.2.*

Since the validated user is non-malicious, their distance vector serves as a good representation for other non-malicious users. Therefore, we estimate the threshold τ with $\theta \times \max(\boldsymbol{v}_T)$, where $\boldsymbol{v}_T \in \mathbb{R}^m$ is the distance vector of the validation user and $\max(\cdot)$ means getting the maximum value of the vector $\boldsymbol{v}_T$. Then, the maximum distance value ($\max(\boldsymbol{v}_j)$) of each selected user will be compared with the threshold (τ) to determine the final list of users who can participate in the update. A user with a maximum distance smaller than the threshold is considered a benign user, while a user with a maximum distance larger than or equal to the threshold will be removed. However, this naive threshold is very unstable as a lucky malicious user can get past it in some rounds due to the instability. Therefore, we make an additional modification, **global-min mean smoothing**, to this basic threshold to address the concern.

Global-Min Mean Smoothing. A naive way to stabilize the threshold value is using a smoothing method, like a moving average. However, in the early communication rounds, the naive threshold value rapidly decreases as the model starts making connections between inputs and output classes. Therefore, applying a traditional smoothing method can result in a relatively high threshold early, which may let attackers bypass it. When the naive threshold (τ) decreases rapidly, we do not wish to use any previous communication rounds for the smoothing.

Therefore, we propose to use the lowest observed value (Global Min) of τ as the starting point of the smoothing window. Let τ_t represent the naive estimation of the threshold in round t. Then the smoothed threshold $\tilde{\tau}$ at round n is given by

$$\tilde{\tau} = \frac{1}{n - t_s + 1} \sum_{t=t_s}^{n} \tau_t,$$

where t_s is the round that when the global min is observed. Details of the global-min mean smoothing are described in Algorithm 2. As τ_t shrinks, we observe new global minimums, and the start of the threshold smoothing is reset. In addition, when our estimates do not continue to fall, previous values are leveraged to smooth the cutoff, which keeps lucky malicious users from getting past a volatile threshold. Figure 3 compares our global min-mean smoothing with the naive threshold and various smoothing techniques. The global min-mean smoothing best captures the naive threshold's early behavior while providing remarkable stability improvements. Most importantly, when our threshold encounters a new global minimum, it provides a conservative estimate to prevent malicious users while re-learning cutoff behavior over the next few rounds.

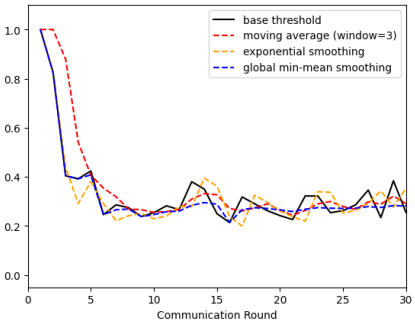


Figure 3: Comparison of the global min-mean smoothing with the base (naive) threshold and various smoothing methods.

Algorithm 2 Global-Min Mean Smoothing
Notation: Let $(\tau_1, \cdots, \tau_{n-1}, \tau_n)$ denote the sequence of values that we wish to smooth.

1: **procedure** GLOBAL-MIN MEAN SMOOTHING($\tau_1, \cdots, \tau_{n-1}, \tau_n$)
2: Record the location of global minimum: $i = \arg\min_{t \in [1,\ldots,n]} \tau_t$
3: Subset to a sequence starting with the global min: $\{\tau_t\}_{t=i}^{n} = \{\tau_i, \cdots, \tau_n\}$
4: **return** average of sequence subset, $\overline{\{\tau_t\}_{t=i}^{n}}$
5: **end procedure**

4 EXPERIMENTS

4.1 SETTING

Federated Learning. We start by giving further specifications regarding the federated learning environment. Our interest is training a global model over M communication rounds with N users. Each iteration randomly selects K users, using a specified proportion of the total users, to participate

in the model update. After local training, the next global model is the average returned model weights by the FedAvg procedure. We use ResNet18, a popular classifier proposed in He et al. (2016), in all our experiments. We assume that all users, including malicious, have complete control over all aspects of local training, such as learning rate, the number of epochs, and the model weights they return. For simplicity, we select two main sets of training hyper-parameters for benign and malicious users. The malicious users will poison a given proportion of their local data by adding their backdoor trigger to the input and changing the training label to the target class. They intend for the model to associate the trigger with the target class and hence have the future global model identify any input with the trigger as belonging to the target class.

Attack and Baseline. To show the effectiveness of our method, we choose a setting in which the backdoor attack is strong. We force all malicious users to be included in the subset of selected users to update the global model each round after the start of the backdoor attack. Note that the selection of random users is a defense against malicious users by making it difficult for them to update the global model repeatedly. Additionally, we do not allow the validation user, a guaranteed benign user, to participate in any global model updates. We make these decisions to highlight the ability of our threshold to prevent even unreasonably strong backdoor attacks against the global model. For our experiment, we test the proposed method and two other robust aggregation methods, Median and Trim-mean (Yin et al., 2018a), against two state-of-the-art backdoor attacks in FL: Neurotoxin (Zhang et al., 2022) and Distributed Backdoor Attacks (DBA) (Xie et al., 2020). To further evaluate the effectiveness of the aggregation methods, we also vary the proportion of malicious attackers to comprise between 10% and 40% of the selected users to test the defense methods under different attack strength levels.

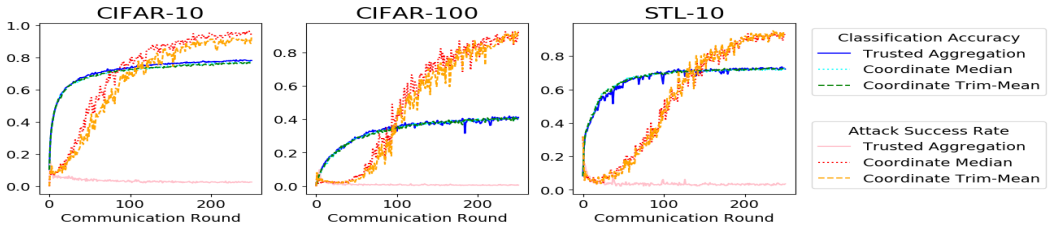
Data. The experiments are done on three different data sets: CIFAR10 (Krizhevsky & Hinton, 2009), STL10 (Coates et al., 2011) and CIFAR100 (Krizhevsky & Hinton, 2009). In each experiment, we construct user data sets by random sampling from the training data. For global model evaluation, we split the test set into two parts. We add the backdoor trigger to images in the second half and remove any target class observations. We measure model performance on the first half using classification accuracy, and the proportion of the poisoned half predicted as the target class, known as attack success rate, to measure the extent that the backdoor attack has compromised the model. For a defense method, a good performance consists of a low attack success rate and high classification accuracy. In other words, attacks are unsuccessful when the defense method is used, and the defense does not negatively influence the classification performance.

4.2 Main Results

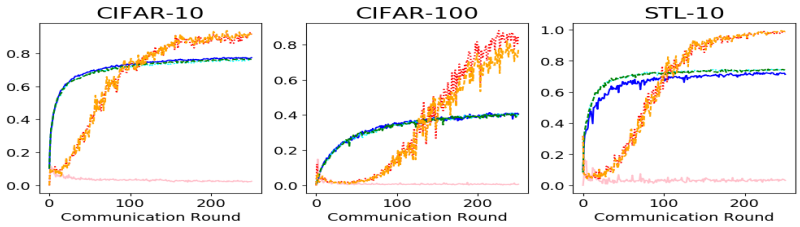
We begin by considering a setting where 10% of the selected users are malicious each communication round. Figure 4 shows the performance of the three robust aggregation methods against backdoor attacks with and without the Neurotoxin projection on three data sets regarding model classification accuracy and attack success rate. For our main results, we use the TAG scaling coefficients (θ's) of 2, 2, and 1.1 for the data sets CIFAR10, CIFAR100, and STL10, respectively. Our proposed method nullifies the backdoor attack in each case without decreasing the classification accuracy of the original task in general. We see a slight improvement in classification accuracy for the CIFAR10 data set and a mild decrease in performance on STL10 under the Neurotoxin attack. However, the other two robust aggregation methods, coordinate-wise Median and Trim-mean, fail to prevent backdoor attacks with or without Neurotoxin. We conclude that our method is a clear improvement to the existing robust aggregation methods for federated learning.

We show that TAG can handle even stronger attack settings against state-of-the-art attacks in the following parts. First, we consider testing the robust aggregation methods against DBA and Neurotoxin attacks with 20% malicious users in the selected set. Unsurprisingly, the robust aggregation methods, which failed against the previous setting, cannot prevent stronger attacks, see Figure 5. On the other hand, our method, TAG, again prevents all backdoor attacks.

Then, we attempt to test our proposed method's limitations by considering DBA and Neurotoxin attacks with 40% malicious users in the selected set. These attacks are catastrophically successful against the current robust aggregation methods, see Figure 6, having a nearly perfect attack success rate after round 50 on all our data sets. However, our method, TAG, overcomes the backdoor extent of the initial rounds to prevent the attack against all data sets. Hence, we conclude that even when the federated learning setting updates consist of almost half malicious users, we can use TAG to attempt

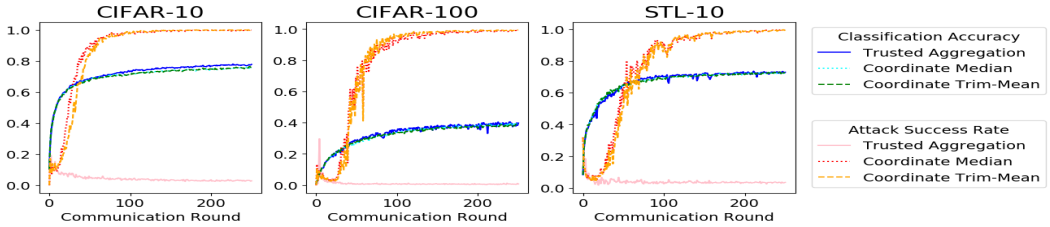


(a) Performance Under Backdoor Attack With 10% Malicious Users.

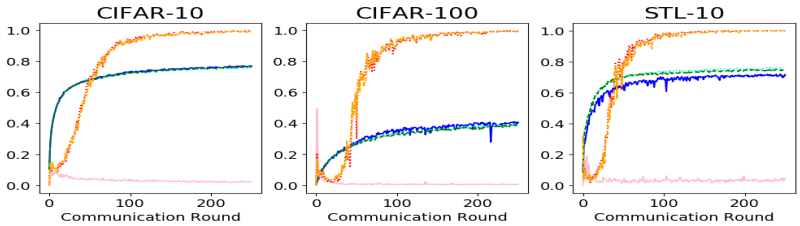


(b) Performance Under Neurotoxin Backdoor Attack With 10% Malicious Users.

Figure 4: Model performance under standard and Neurotoxin backdoor attacks with 10% malicious users. The proposed method TAG performs well in defending against backdoor attacks as the attack success rates are low. Meanwhile, it does not generally affect the model's classification performance on clean data, except for a slight decrease for the Neurotoxin attack on STL10. However, the other two aggregation methods do not work well against any backdoor attacks.



(a) Performance Under Backdoor Attack With 20% Malicious Users.



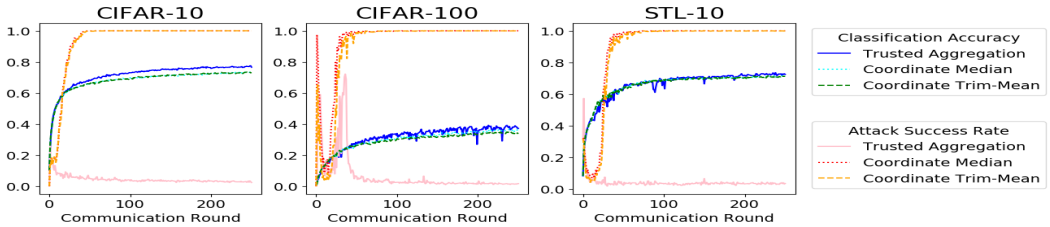
(b) Performance Under Neurotoxin Backdoor Attack With 20% Malicious Users.

Figure 5: Model performance under standard and Neurotoxin backdoor attacks with 20% malicious users. TAG is the only method that prevents these backdoor attacks.

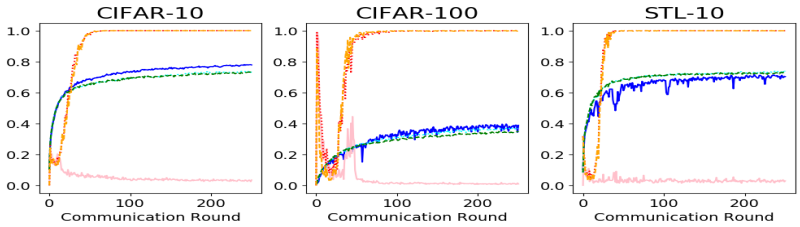
to eliminate backdoor attacks. Additionally, in the Appendix, see Figure 10, we show that our model does not hinder performance for the original classification task without a backdoor attack. Hence our method leads to increased model security at negligible cost to the shared model under a lack of attack. We conclude that Trusted Aggregation (TAG) is an essential advancement toward model security for the federated learning framework.

4.3 Necessity Of Global-Min Mean Smoothing

We revisit the last attack considered on the STL10 data set to highlight the importance of our proposed technique, global-min mean smoothing. Recall that our smoothing is intended to improve the stability of our estimated threshold while preserving its behavior in the initial rounds, not conserved by other smoothing techniques. If we repeat the backdoor attack where 40% of the user subset is malicious each round and omit the global-min mean smoothing, our method no longer can prevent these



(a) Performance Under DBA Backdoor Attack With 40% Malicious Users.



(b) Performance Under DBA And Neurotoxin Backdoor Attacks With 40% Malicious Users.

Figure 6: Model performance under DBA and Neurotoxin backdoor attacks with 40% malicious users. Again, TAG is the only method that can prevent backdoor attacks.

backdoor attacks, see Figure 7. Here our method succumbs to the backdoor attack nearly as quick as the baseline robust aggregation methods. Hence, without global-min mean smoothing, we conclude the malicious users may have a larger opportunity to get past an unstable cutoff.

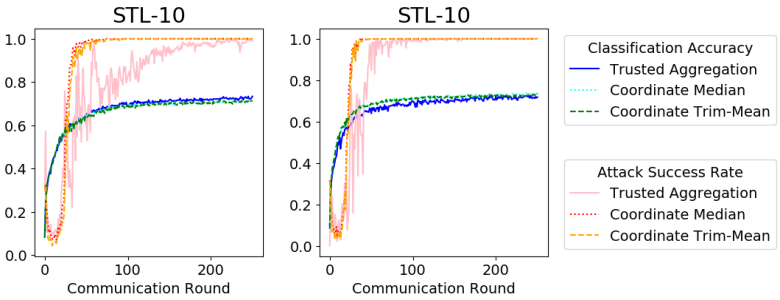


Figure 7: Model performance under DBA without (*left*) and with (*right*) Neurotoxin backdoor attacks with 40% malicious users on STL10 without using global-min mean smoothing. TAG fails to defend against the backdoor attack on STL10 without the improved stability of our estimate from our proposed smoothing technique, global-min mean smoothing.

4.4 Data Distribution Study

In this section, we show the applicability of our method for imbalanced users focusing on the CIFAR10 data set. Recall that local data sets are not required to be independent and identically distributed for federated learning. For our experiments, the user data sets are obtained by random sampling from the training data. In this experiment, we use the m-dimensional (number of total classes) Dirichlet distribution to determine the proportion for each class to be randomly sampled. We parameterize this distribution by the simplified $\alpha \mathbf{1}_m$ where $\mathbf{1}_m$ is an m-dimensional vector of ones, and $\alpha \in [0, \infty)$ is a scalar such that smaller values lead to imbalanced user data.

Regarding attacks with 10% and 20% malicious users, we did not have to modify the scaling parameter θ from previously balanced data experiments to have a very successful defense when data is highly imbalanced ($\alpha = 1$), see results in Figure 11 in the Appendix. Our method prevents such backdoor attacks while achieving great classification accuracy for the original task. This setting alone is more than sufficient to show the usefulness of our proposed method TAG in the majority of federated learning settings.

Furthermore, TAG is successful against the unreasonably strong 40% malicious setting if we reduce the scaling coefficient θ. Recall the trade-off between robustness against backdoor attacks and

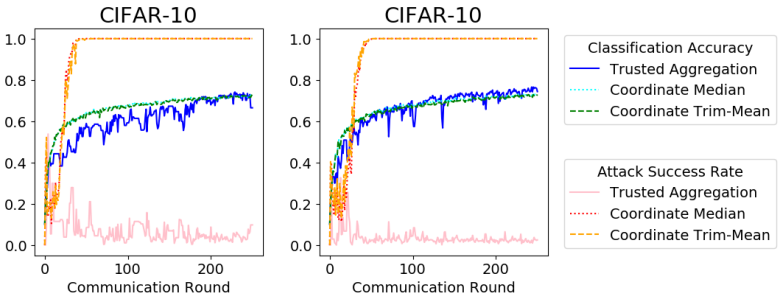


Figure 8: Model performance under DBA without (*left*) and with (*right*) Neurotoxin backdoor attacks with 40% malicious users on imbalanced local user data sets. The proposed method, TAG, performs well in defending against backdoor attacks, even when the user data is randomly sampled using the Dirichlet distribution with $\alpha = 1$. However, for the extreme case with 40% malicious, the scaling parameter for TAG, θ, needed to be reduced to prevent stronger attacks. Again, the other two aggregation methods do not work well in preventing any backdoor attack under imbalanced data.

performance for the original classification task. In Figure 8, with $\theta = 1.25$ and $\theta = 1.5$ respectively, TAG is able to eliminate the backdoor attacks. With smaller scaling, the original classification accuracy suffers in early communication rounds. However, as the global model nears convergence, the round-to-round accuracy stabilizes and becomes equal to or better than the baseline robust aggregation methods. Hence, we conclude that our proposed method, TAG, can handle backdoor attacks on reasonably imbalanced data without losing accuracy for the original task. These experiments support our recommendation to choose the smallest scaling possible that achieves desired performance on the original task to prevent backdoor attacks.

4.5 Validation User Data Proportion Study

For previous experiments, we have assumed that the validation user has a data set of the same size as the other users. In most cases, data can be acquired. However, we want to determine whether our method depends on such size requirements. Hence we revisit the strongest attack setting for the CIFAR10 data set but only allow the validation user to have a data set that is 20% of the size of the other local users. Here the validation set is now only allowed 100 images, yet TAG prevents the backdoor attack and can achieve improved accuracy compared to the baseline robust aggregation methods. Hence, we conclude we do not need validation data of the same quantity as other local users to discriminate between benign and malicious returning models.

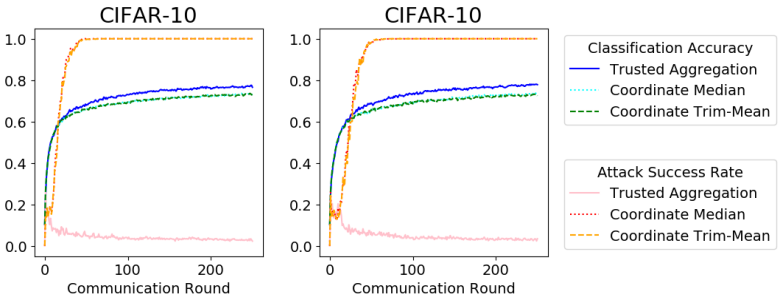


Figure 9: Model performance under DBA and Neurotoxin backdoor attacks with 40% malicious users on CIFAR10 where the validation data set is 20% the size of the local users. Still, the proposed method TAG performs well, and the other two aggregation methods do not work well in preventing any backdoor attack.

5 Conclusion

While current robust aggregation methods fail to prevent mild backdoor attacks, TAG holds up against state-of-the-art attacks in unreasonably strong settings. Furthermore, TAG works when users have heterogeneous and imbalanced data, does not decrease the classification accuracy for the original task, and only a minimal validation set is required. Finally, TAG is compatible with other filtering methods or modifications to the choice of aggregation step. We conclude that TAG is an essential advancement toward model security for the federated learning framework.

REFERENCES

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223, 2011.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. pp. 2793–2806, July 2020. doi: 10.18653/v1/2020.acl-main.249. URL https://aclanthology.org/2020.acl-main.249.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. Backdoor attacks on pre-trained models by layerwise weight poisoning. pp. 3023–3032, November 2021. doi: 10.18653/v1/2021.emnlp-main.241. URL https://aclanthology.org/2021.emnlp-main.241.

Dhurgham Hassan Mahlool and Mohammed Hamzah Abed. A comprehensive survey on federated learning: Concept and applications. 2022. doi: 10.48550/ARXIV.2201.09384. URL https://arxiv.org/abs/2201.09384.

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. 2016. doi: 10.48550/ARXIV.1602.05629. URL https://arxiv.org/abs/1602.05629.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Onion: A simple and effective defense against textual backdoor attacks. *arXiv preprint arXiv:2011.10369*, 2020.

Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1354–1371. IEEE, 2022.

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

Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgyS0VFvr.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5650–5659. PMLR, 10–15 Jul 2018a. URL https://proceedings.mlr.press/v80/yin18a.html.

Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5650–5659. PMLR, 2018b.

Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26429–26446. PMLR, 17–23 Jul 2022.

A SUPPLEMENTARY RESULTS

A.1 MODEL PERFORMANCE WHEN NO BACKDOOR ATTACK

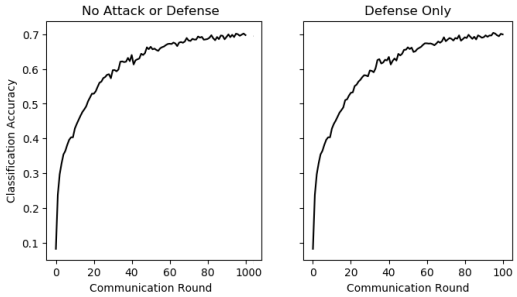
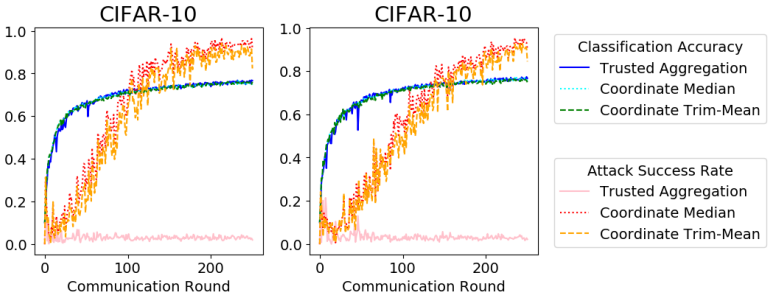
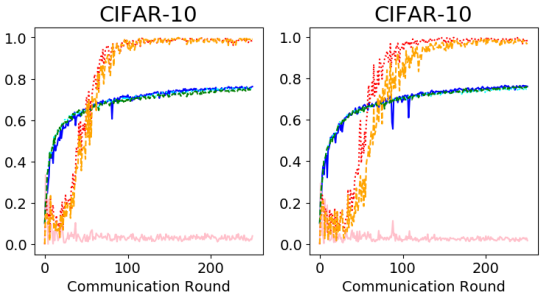


Figure 10: Model classification accuracy on STL10 in the absence of backdoor attacks without (*left*) and with (*right*) TAG defense. In addition to successfully preventing attacks when present, we observe that our defense does not hinder the classification accuracy of the original task for the FedAvg procedure. This supports the overall usefulness of our proposed method as the shared model will have improved security with our defense without cost if the system is not threatened by attacks.

A.2 ADDITIONAL FIGURES FOR DATA DISTRIBUTION STUDY



(a) Imbalanced Data Performance Under Backdoor Attack Without (*left*) And With (*right*) Neurotoxin Backdoor Attacks With 10% Malicious Users.



(b) Imbalanced Data Performance Under Backdoor Attack Without (*left*) And With (*right*) Neurotoxin Backdoor Attacks With 20% Malicious Users.

Figure 11: Model performance under DBA without (*left*) and with (*right*) Neurotoxin backdoor attacks with 10% and 20% malicious users on imbalanced local user data sets. The proposed method, TAG, performs well in defending against these backdoor attacks, even when the user data is randomly sampled using the Dirichlet distribution with $\alpha = 1$ without changing the scaling parameter (θ). Again, the other two aggregation methods do not work well in preventing any backdoor attack under imbalanced data.

B SUPPLEMENTARY PROOFS

B.1 PROOF OF PROPOSITION 1

$$\text{Assume } \forall c \in [1, ..., m], v^{(c)} \sim \text{ Unif}(0, b_c)$$

$$\text{Define } V = \max_c \left(v^{(c)}\right) \text{ and } j = \arg\max_c (b_c)$$

$$v^{(j)} \leq V \leq b_j \implies E\left[v^{(j)}\right] = \frac{b_j}{2} \leq E\left[V\right] \leq b_j \implies E\left[V\right] \leq b_j \leq E\left[2V\right]$$

B.2 PROOF OF PROPOSITION 2

$$\text{Assume } \forall c \in [1, ..., m], v^{(c)} \overset{\text{ind}}{\sim} \text{ Unif}(0, b_j)$$

$$\text{Let } W = \max_c \left(w^{(c)}\right) \text{ where } w^{(c)} \overset{\text{iid}}{\sim} \text{ Unif}(0, b_j) \implies \frac{W}{b_j} \sim \text{Beta}(m, 1) \implies E(W) = \frac{m}{m+1}$$

$$F_V(t) = P(V \leq t) = \prod_{i=1}^{m} P_{v^{(c)}}(t) \geq \prod_{i=1}^{m} P_{w^{(c)}}(t) = P(W \leq t) = F_W(t)$$

$$\implies E(V) = \int_0^\infty (1 - F_V(t))\, dt \leq \int_0^\infty (1 - F_W(t))\, dt = E(W)$$

$$\implies E\left[\frac{V}{b_j}\right] \leq E\left[\frac{W}{b_j}\right] = \frac{m}{m+1} \implies E\left[\left(\frac{m+1}{m}\right) V\right] \leq b_j$$