Optimal Transport Based Dynamic Weighted Federated Aggregation to Mitigate Poisoning Attacks

Anonymous CVPR submission

Paper ID 9680

Abstract

1. Introduction

Multi-agent federated learning (FL) has attracted the attention of researchers because of its use in several applications, such as signal processing, mobile user personalization, speech recognition, etc. [65], [20, 38]. Federated learning can simultaneously offload the computation and memory-intensive training work onto multiple low-end computation devices, referred to as clients [21, 22]. Evolving from the early works about query processing in multi-agent signal processing, sensor networks, edge, and fog computing [19, 43], FL functions as a highly distributed decentralized system preserving data privacy [79] with limited communication and computational capabilities [28]. The clients (or agents) train the local model based on collected (sensor) data. The server receives regular analog transmitted signals (updates) of the local models trained at the clients. It then builds the *global shared model* from these updates and communicates the new weights back to the clients. This *iterative learning* is synchronized over several messages to produce a high-quality, stable global model [34, 46]. This process makes FL a more privacy-friendly approach instead of the traditional approach based on centralized data collection and model training [11].

Despite the above advantages of shared intelligence, FL with deep neural networks (DNN) faces many unique challenges, such as shared model training, collective vs. minimal updates, effective aggregation techniques, non-uniform data distribution, computational and memory constraints [45, 62]. Also, there are many security and privacy concerns about data provenance [64, 77], the trustworthiness of sources, and overall confidentiality [29], [11, 50]. More importantly, despite the reputation for privacy-preserving properties about secured aggregation [34, 41], FL is found to be vulnerable to poisoning attacks [32, 71]. Poisoning comes in two forms, namely, data Poisoning and model Poisoning. Contrary to data poisoning, malicious clients modify the received model by tampering with its gradient/parameters before sending it back to the central server for aggregation. As a result, the global model can be severely poisoned with invalid gradients during the aggregation process.

Past works [9], [55, 84] have exposed its high vulnerability to minor perturbations of adversarial attacks under the *white-box setting*. However, they are somewhat unrealistic in an FL setting as the attacker needs to have complete knowledge of the model structure and parameters distributed over all the clients. The white-box attack becomes unfeasible in a real-world scenario given the multiple levels of security, including data encryptions to hide the model details. In contrast, the adversary trains the local model using malicious data without much knowledge in the *black-box adversarial attacks*. The need for distributed client updates and model training in the presence of malicious clients in FL increases the vulnerability points for data leakage. *This distributed system issue of expanding FL attack surface was not relevant for the centralized scenario*. Hence, it was not part of prior attack research focused on traditional deep learning. Our state-of-art study has shown that the FL research community is showing interest in investigating black-box adversarial attacks [50] due to the debilitating damage. After a few rounds of updates, the adversarial attack in FL forces the learning algorithm in the global server model to cause drastic misclassification with extremely high confidence. The FL setting becomes relevant in autonomous vehicles (AV) as each automobile can be treated as an individual client and demonstrate the drastic mispredictions in traffic sign recognition due to such attacks. This can lead to a gross misprediction of crucial elements and eventually serious safety and security issues. In the larger context, these malicious attacks may prevent the widespread adoption of AV due to crashes with other vehicles or pedestrians. In this paper, *we focus on untargeted data poisoning attacks* in FL as it is the most common and relevant to production deployments [60]. Specifically, the attacker is interested in *generic misclassification (untar-*

geted) rather than specific misclassification (targetted).

2. Related Work

This section reviews existing literature and relates to our proposed methodology from two perspectives: adversarial attacks in federated learning and cryptography-based defense methods in machine learning.

2.1. Adversarial Attacks

These attacks [14, 73] have received immense popularity in the deep learning research community as it causes debilitating accuracy failures in autonomous vehicles, NLP, speech recognition, and other safety-critical applications, among others. Barely perceptible to the human eye, these attacks were first observed in the seminal articles [66], [27]. Recent comprehensive review articles [3, 53], [39] have highlighted the taxonomy of white and black-box adversarial attacks, including their fundamental differences. Using multiple queries to build the surrogate iteratively, the adversary utilizes its transferability properties to craft the adversarial model [49, 52], [15]. Giorgos *et al.* [68] proposed a targeted adversarial mismatch attack that generates an adversarial image at query time for deep learning-based image retrieval systems. Sanli *et al.* [67] proposed Type I adversarial attack to cheat image classifiers by designing a supervised variational autoencoder. In early works [49, 52], [15], the attacker first builds the surrogate model by sending queries (of the order of millions) to learn about the original model and eventually transfers it to the final attack. According to the dichotomy of attacks presented in the recent articles [51, 81], iterative methods are considered most practical for black-box attacks in image classification problems. However, all these attacks suffer from the issue of a large number of queries and late convergence. Hence, we use Modified Simple Black-box Attack (M-SimBA) proposed by Kumar *et al.* [36] which tries to minimize the loss of most confusing class probability with fewer iterations and generate robust adversarial samples.

Adversarial attacks can be broadly classified into two categories: causative/poisoning (training time) attacks and evasion/exploratory (test time) attacks [82]. Evasion attacks are also known as "adversarial examples", in which crafted undetectable perturbations are appended to the test data. Recent works [9, 84], [55] have shown that FL is highly vulnerable to minor perturbations of adversarial attacks under the white-box setting. Xie *et al.* [78] introduced the idea of distributed backdoor attacks in the FL framework, in which the dishonest participants in FL add local triggers to their training data. Finally, they try to influence the global model to classify triggered images in the desired way.

Melis *et al.* [47] demonstrate that local model updates leak unintended information about participants' training data and develop passive and active inference attacks in order to exploit this leakage. They also show that an adversarial participant can infer the presence of exact data points such as specific locations in others' training data using the leaked information. While these are a few exciting approaches for adversarial attacks, the *black-box evasion adversarial attack in FL using gradient perturbation* has been a barely explored topic in the literature. Hence, contrary to all these methods, we focus on defending FL against black-box evasion attacks at the central server.

2.2. Optimal Transport in Machine Learning

Optimal Transport (OT) theory has been gaining significant attention from the machine learning community due to its efficiency in modeling various ML applications [70]. *Computer vision:* early works [56] used OT formulations (Wasserstein distance) in computer vision applications to find the dissimilarity measure between the images. Specifically, an effort has been made for content-based image retrieval by investigating the properties of the earth mover's distance (EMD). Further, OT is used to perform the image-to-image color transfer, the color of a source image to match the color of a target image of the same scene [4, 54]. *GANs:* research has been done to improve generative adversarial networks (GANs) using OT [1, 5, 58]. Liu *et al.* [40] proposed WGAN-QC, a WGAN with quadratic transport cost (Optimal Transport Regularizer) to stabilize the training process of WGAN-QC and prove that it converges to a local equilibrium point with finite discriminator updates per generator update. *Semantic correspondence:* Liu *et al.* [42] tries to establish dense correspondences across semantically similar images by solving the many-to-one matching and background-matching issues using OT. *Domain adaptation:* early works [16, 61] introduced a regularized optimal transport model in an unsupervised way to align the representations in the source and target domains. Flamary *et al.* [24] solve domain adaptation problem by learning a transportation plan from the source domain to the target domain. Other applications include the generative model. *Graph matching:* Gromov *et al.* [80] proposed a novel Gromov-Wasserstein learning framework to jointly match (align) graphs and learn embedding vectors for the associated graph nodes. Finally, very few works used OT to improve the federated learning system [23, 75]. However, to the best of our knowledge, there is no explicit use of OT in FL to defend against data-poisoning attacks. We are the first to model a defense mechanism using the OT framework.

3. Proposed Approach

3.1. Overview of Federated Learning

Federated Learning involves bringing machine learning (ML) capabilities to local clients for building models with local datasets, ensuring their privacy. It consists of a total

N clients, each with access to local data D_i, where i is the client's index, $i \in N$ and $1 \leq i \leq N$. Each client maintains its own copy of the data shard as private, such that $D_i = \{x_1^i ... x_{l_i}^i\}$ and $|D_i| = l_i$ is not shared with the server.

Initialization:: The server generates the initial global model by training on some amount of auxiliary data.

Client execution: At each global epoch t, every client: (i) tries to minimize the empirical loss over its data shard and trains the classification algorithm with a batch size of B for E local epochs with the initial global model w_G^t, (ii) after the completion of training phase with E local epochs, all client(s) calculate the local update using $\Delta C_{t+1}^n = w_{t+1}^n - w_t^G$, and (iii) these individual client model updates are sent back to the central server for model aggregation. We consider two FL settings, namely, (i) homogeneous with equal data size and (ii) heterogeneous with different data sizes among connected clients.

Server execution: At each global epoch t, the central server: (i) sends the current version of the global model to update all N agents, (ii) receives the local client updates and performs global model aggregation using synchronous federated weighted averaging as $w_{t+1}^G = w_t^G + \sum_{n \in N} \alpha_n \Delta C_{t+1}^n$, where, $\alpha_n = \frac{l_n}{\sum l_n}$ and $\sum_n \alpha_n = 1$ i.e., the updates are used to generate the 'aggregated' global model synchronously for T global epochs, and (iii) performs global model testing using the updated global model on the test data at the server. Further, federated weighted averaging is a naive aggregation rule that averages the local model parameters to obtain the global model parameters. It is widely used under non-adversarial settings [18,45]. However, federated averaging is not robust under adversarial settings as the an attacker can manipulate the global model parameters arbitrarily for this mean aggregation rule when compromising only one worker device [10, 82]. Hence, we take optimal transport based approach to improve upon federated averaging and mitigate data poisoning attack in FL.

We ensure a *non-i.i.d* (independent and identically distributed) dataset by splitting the dataset randomly among clients with the number of samples as $l_k >= \lambda$, where λ is the minimum threshold to ensure valid FL protocol. The central server's objective is to learn a global weight parameter w_G. It uses iterative aggregation of model updates from the clients bounded by n-dimension parameter space and then evaluates on the remaining test data.

3.2. Threat Model

We discuss the configuration details for the threat model w.r.t. client and server, along with adversary goals and capabilities. Our threat model consists of single (fixed attacker clients) & multi-client attack (random malicious clients) scenarios. The global model is a image classifier in our paper. The threat model considers that the malicious client(s) can 'silently' poison the global model with malicious messages disguised as a benign participant. This implies that the adversary can poison the central server only through the local model update, poisoned using malicious data. Avoiding a single point of failure, the aggregation algorithm is considered beyond the attacker's control. In addition, malicious clients cannot directly poison other benign participants' learning phase or training data $\mathbb{D}_{benign}$, until and unless the node is explicitly specified as an adversary node. This implies that clients marked malicious must train on their respective poisoned dataset $\mathbb{D}_{poison}$ as per federated learning.

Adversary goals: Our adversary performs causative black-box adversarial attack in a federated learning setup with (i) minimalistic information about the model architecture and parameters, (ii) can access only the predicted class labels and probability scores. The malicious client(s) (adversary) aims to poison the global model by perturbing the local training data disguised as a benign participant. The adversary uses black-box method to create the adversarial samples. The perturbed local dataset is used to train the local model to generate a poisoned local update. After the federated averaging at the server-side, the global model shows mis-predictions on the test data.

Adversary capabilities: The adversary capabilities are (a) *Adversarial sample generation* - the attacker can sniff and modify a portion (or entire) local data shard to corrupt the training procedure. The adversary has 'limited' access control over the local clients and operates covertly to remain undetected. It implies that the central server ensures the completion of training cycles and the exchange of updates in a scheduled manner. (b) *Uninformed* - the local model is a black box for the attacker. It implies that the attacker has minimal knowledge about the internal architecture and gradients. However, the adversary can infer the model behavior to effectively generate random gradient perturbations.

3.3. Adversarial Attacks in Federated Learning

Adversarial attacks against ML models and DNN have received much attention [13,27,57]. There has been tremendous research interest in designing novel adversarial attacks for deep neural networks because of its gross mispredictions, even with minor perturbations [27]. This interest has percolated into the privacy-preserving federated learning [44, 85] as researchers have begun investigating it through the lens of adversarial settings [9]. Given that clients in FL communicate by providing data and local model updates to the central server, adversarial attacks in FL are usually performed either through client data (data poisoning) or model updates (model poisoning). Broadly, there are two major types of adversarial attacks in FL: targeted & untargeted data poisoning and model poisoning. Below, we discuss the existing works in different kinds of attacks that are designed to attack FL.

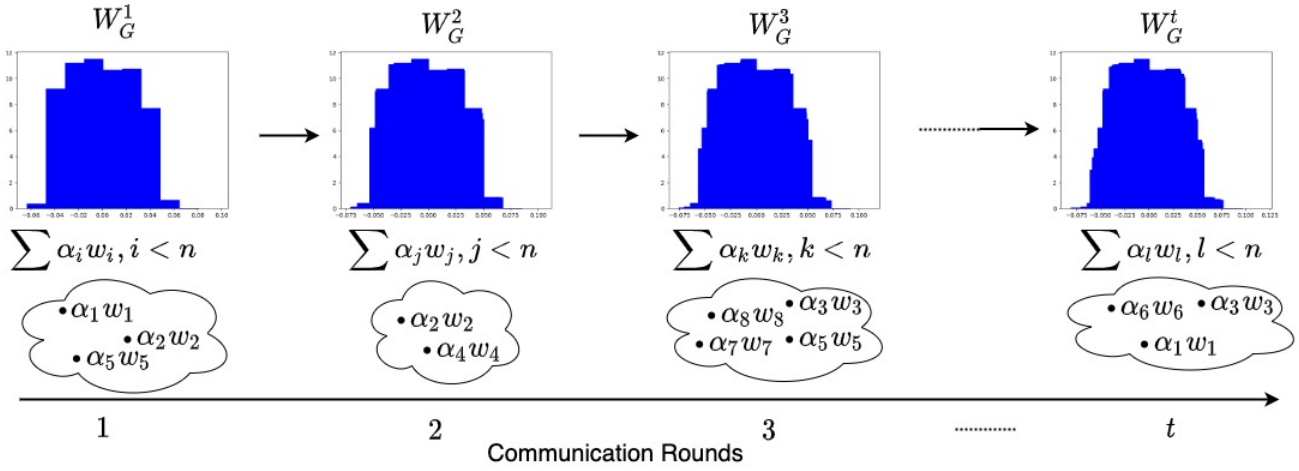

Figure 1. ..

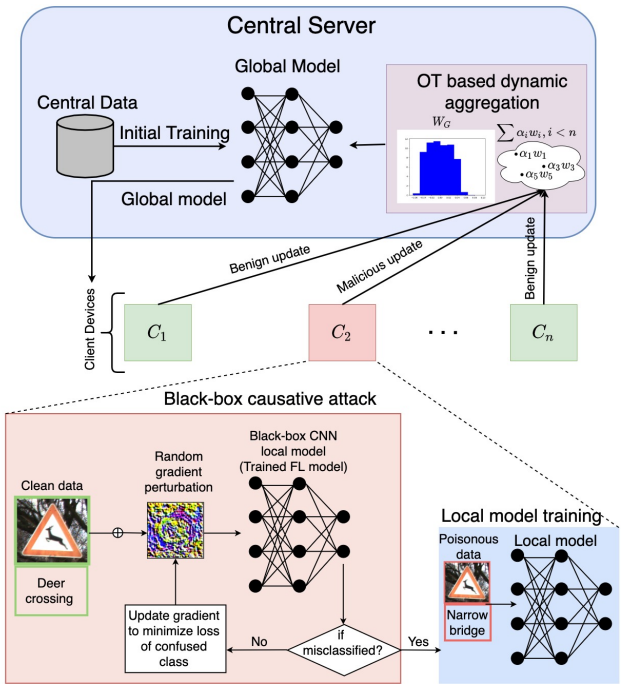

Figure 2. ..

Data Poisoning Attacks: In this category, the attacker or the malicious client tries creating data (i.e., poisonous) that - through local model updates - leads to an incorrect or imprecise global model. The black-box malicious attack in multi-agent communication is first introduced in [72] using a computationally expensive surrogate-based approach. Zhang *et al.* [83, 84], Hitaj *et al.* [31], Wang *et al.* [76] proposed a generative adversarial attack (GAN) based poisoning attack method in the context of federated learning. Tolpegin *et al.* [69] studied targeted data poisoning attacks against FL systems in which a malicious subset of the participants aim to poison the global model by sending model updates derived from mislabeled data. Wang *et al.* [74] proposed an edge-case backdoor attack that forces a model to misclassify on seemingly easy inputs that are, however, unlikely to be part of the training or test data and live on the tail of the input distribution.

Model Poisoning Attacks: In this second category, the attacker directly sends malicious updates [8, 9]. Bhagoji *et al.* [8, 9] have focused on targeted model poisoning as opposed to data poisoning of prior works. Bagdasaryan *et al.* [6] proposed a backdoor FL attack framework that trains on the backdoor data using a constrain-and-scale technique and submits the resulting corrupted model as an update to the central server. Shejwalkar *et al.* [59] proposed a general model poisoning attack by computing the malicious model update through maximally perturbing the benign reference aggregate in the malicious direction.

In this paper, we focus on defense untargeted data poisoning in FL as we find it to be significantly most common and relevant to production deployments [60]. Also, data poisoning attacks can be used to impact a large population of FL clients and remain undetected for a long duration.

3.4. Designing the Adversarial Attack

The attacker performs the following steps for attacking the FL setup: (i) generate adversarial samples based on the gradient based black-box attack method, (ii) add these samples to the local training dataset, (iii) train the local model, (iv) finally transmit the malicious messages as analog signals to the central server to mislead the global model and perform misprediction on test data.

Black-box attack method: In this paper, we consider the

M-SimBA data poisoning attack proposed by Kumar *et al.* [37]. Firstly, to calculate the adversarial image, a random gradient perturbation is added to the original image and is calculated as

$$I_{adv} = I_x + \epsilon * G_p, \tag{1}$$

where I_{adv} is the adversarial image, I_x is the original image, and G_p is the randomized gradient perturbation. The step size (ϵ) controls the intensity of perturbation. The local black-box model uses I_{adv} to calculate the most confused class score, y' as

$$y' =_{\hat{Y} \neq Y} \{P(\hat{Y}|I_x)\}, \tag{2}$$

where Y, $\hat{Y}$ are original and predicted class, respectively.

The algorithm repeats the above process until it generates the final adversarial image as per Eq. (1). For the initial iteration, the gradient is added in the positive direction. The gradient is updated in the negative direction for further iterations and is changed to random perturbations in subsequent iterations. The iterative method creates an adversarial image that will eventually get misclassified. In addition, it converges on the L2 norm such that $||I_{adv} - I_x||_2 < \zeta$. Threshold parameter ζ controls the deviation of adversarial image w.r.t. original image *without* making it perceivable to the human eye.

In the final step, converged gradient perturbation (G_p) is added to the input image according to Eq. (1). This step returns the final adversarial image to the local dataset for further local model training. Further, the local model is trained on generated adversarial data using the local model. The malicious update is uploaded to the central server. Finally, the server performs model aggregation using federated weighted averaging. In this manner, we design a novel black-box attack framework in FL for traffic sign recognition in autonomous vehicles that can effectively generate adversarial samples without the knowledge of the local model.

3.5. Proposed Defense Mechanism

In this subsection, we discuss the background of optimal transport (OT), problem formulation, and proposed algorithm for dynamic model aggregation to discard the poisonous model updates.

Overview of optimal transport (OT): Gaspard Monge introduced OT [48], [33] to find the most efficient way to move a unit of mass between two distributions. The aim is to minimize the overall ground cost to move the unit mass from source distribution to the target distribution. The optimization problem can be given as

$$\min_{t,\, t \neq \mu_s = \mu_t} \int C(a, t(a))\, d\mu_s(a),$$

where μ_s, μ_t correspond to source and target distributions, respectively. $C(.,.)$ is the ground cost of moving a unit mass between two positions x, $t(x)$. The constraint $t \neq \mu_s = \mu_t$ is to ensure that source is completely transported to target. In general, OT solution is used in two main aspects, (i) to find the optimal value that measures the similarity between two distributions, also known as Wasserstein distance. (ii) To find the OT matrix which is the optimal correspondence mapping between distributions.

Wasserstein Barycenters [2]: It is a distribution that minimizes the weighted sum of Wasserstein distance w.r.t all other distributions. It aims to find a distribution μ such that

$$\min_{\mu} \sum_{n} \alpha_n \mathbb{W}(\mu, \mu_n), \tag{3}$$

where $\mathbb{W}(.,.)$ correspond to Wasserstein distance between distributions, α_i respresent the weight of distribution μ_i.

Problem formulation: Let us assume we are at t^{th} communication round in federated learning such that the server receives the model updates from all the n clients, and $\mathbb{D}_v$ is the validation data at the server. Let $\{w_1^t, w_2^t, \ldots, w_n^t\}$ are model updates that correspond to $\{c_1, c_2, \ldots, c_n\}$ clients, respectively. Also, let us assume there are k unknown malicious client updates $k \in n$ based on the attack percentage A_p (the number of malicious clients for the communication round). Now, the aim is to find a global model weight w_G^t that minimizes its weighted Wasserstein distance w.r.t other benign client model weights $w_{[1,2,\ldots,n]}$ after dynamically discarding the malicious updates.

Different variations of OT optimization: Recent developments in OT have resulted in different variations of OT optimization. *(i) Regularized OT:* It is expressed as

$$\gamma^* = argmin_{\gamma \in \mathbb{R}_+^{mxn}} \sum_{i,j} \gamma i, j \mathbb{M}_{i,j} + \lambda \omega(\gamma),$$

$$s.t. \gamma 1 = a; \gamma^T 1 = b; \gamma \geq 0,$$

where $\mathbb{M} \in \mathbb{R}_+^{mxn}$ is the cost matrix to move mass from bin a_i to bin b_j, a, b are histrograms that represent the weight of each samples in the source and target distributions. ω is the regularization term. *(i) Entropic Regularized OT:* Marco Cuturi [17] smooth the classic OT problem with an entropic regularization term, and show that the resulting optimum is also a distance which can be computed through Sinkhorn's matrix scaling algorithm faster than that of transport solvers. It is expressed as $\gamma_\lambda^* = diag(u)\mathbb{K}diag(v)$, where u, v are vectors and $K = exp(-M/\lambda)$ and exp is taken componentwise. In addition, there are other regularization such as *quadratic* ($\omega(\gamma) = \sum_{i,j} \gamma_{i,j}^2$), that has similar effect to entropic regularization yet keeps some sort of sparsity that is lost when $\lambda > 0$ [12]. *Group lasso regularization* given by ($\omega(\gamma) = \sum_{j, G \in \zeta} ||\gamma_{G,\zeta}||_q^p$), where ζ contains non-overlapping groups of lines in the OT matrix [24]. Further, there are other optimizations and problem formula-

tions such as Wasserstein discriminant analysis [25], unbalance OT [26], etc. Finally, our proposed optimization is intuition-based with respect defending against data poisoning attacks in FL. We formulate our problem statement in terms of Wasserstein barycenter as per Eq. 3.

Proposed optimization: We introduce FLOT, an OT based byzantine-resilient dynamic weighted federated aggregation rule to mitigate poisoning attacks. *Byzantine Resilience of Proposed Optimization*. Blanchard *et al.* [10] prove that no linear combination of the vectors can tolerate a single byzantine worker. Specifically, federated averaging [45] is not byzantine resilient. Further, they define the conditions on an aggregation rule to be byzantine-resilient. Intuitively, the aggregation rule should output a vector $\mathbb{V}$ that is not too far from the *real* gradient G. In addition, existing byzantine robust algorithms algorithms like KRUM [10] select the local model updates that are representative of a majority of the client models by computing the pairwise distances between individual models. However, when the data across the workers is highly non-iid, there is no 'representative' client model. The local client models show high variance with respect to each other as they compute their local gradient over vastly different local data. Hence, for convergence it is important to not only select a good (non-Byzantine) local model, but also ensure that each of the good models is selected with roughly equal frequency. Further, KRUM when applied to non-iid datasets performs poorly even without any attack [30]. This is because KRUM mostly selects models from $n-c-2$, (where c is the number of malicious clients), local models whose pairwise distances is closer to others. Hence, the robust aggregation rules may fail on realistic non-iid datasets.

To circumvent this issue, we consider loss function based rejection (LFR) with OT optimization to develop Wasserstein barycentric aggregation rule (FLOT). At the end, through our experimental results, we show that, our FLOT also serves as robust client selection technique in discarding the benign clients that does not perform well on the validation data. This implies that dropping some less performing benign updates helps to improve the accuracy, which also supports the claims of the recent work, DivFL [7].

Now, we explain our FLOT framework. To start with, we find the optimal coefficients set of the client model weights α based on the validation loss $\mathcal{L}_v$ of every client model w_i. It can be formulated as

$$\alpha \leftarrow \mathcal{L}_v(w, \mathbb{D}_v) \tag{4}$$

$$\alpha' \leftarrow |\alpha - \max(\alpha)| \tag{5}$$

Now, we define a set $\alpha'_0 = \alpha'$ and write

$$\beta_1 := \{b \in \alpha'_0 : b \leq a \, \forall \, a \in \alpha'_0\}. \tag{6}$$

Next, we define $\alpha'_1 := \alpha'_0 \setminus \beta_1$ which discards the highly malicious weight coefficient from the set α'_0. Further, we inductively write

$$\beta_k := \{b \in \alpha'_{k-1} : b \leq a \, \forall \, a \in \alpha'_{k-1}\}, \tag{7}$$

$$\alpha'_k := \alpha'_{k-1} \setminus \beta_k \tag{8}$$

such that α'_k is the final set after discarding k malicious client updates. Further, we normalize α'_k to $[0, 1]$ through the softmax of all weighting factors, which is defined as:

$$\alpha'_k = \frac{e^{\alpha'_k}}{\sum_{k=1}^{n} e^{\alpha'_k}}. \tag{9}$$

Now, our optimization problem can be formulated in terms of Wasserstein barycenter as per Eq. 3 as

$$FLOT(w_1^t, w_2^t, \ldots, w_n^t) \leftarrow \min_{w_G^t} \sum_k \alpha'_k \mathbb{W}(w_G^t, w_k) \tag{10}$$

Lemma 1. *The expected time complexity of our FLOT function $FLOT(w_1^t, w_2^t, \ldots, w_n^t)$, where, $w_1^t, w_2^t, \ldots, w_n^t$ are d-dimensional vectors is $\mathcal{O}(n.d)$.*

Proof. Firstly, the parameter server computes the maximum of loss values $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ and updates all its elements $|\alpha - max(\alpha)|$ (time $\mathcal{O}(n.d)$). Then the server selects the loss that are less than a certain threshold (expected time $\mathcal{O}(nlog(n).d)$ with binary search). Next, it computes the set difference to discard the highly malicious weight vector (time $\mathcal{O}(n.d)$). Finally, the server normalizes the remaining $n-k$ values (time $\mathcal{O}(n.d)$). Hence, adding all the times we obtain the overall time complexity of $FLOT$ as $\mathcal{O}(n.d)$.

We report that **our proposed FLOT time complexity is $\mathcal{O}(n.d)$ which is a significant improvement over $\mathcal{O}(n^2.d)$ of the Krum function [10].**

4. Experiments

4.1. Datasets

We demonstrate the efficacy of the proposed FLOT method on three benchmark datasets, namely, the German traffic sign recognition benchmark (GTSRB) dataset [63], KUL Belgium traffic sign (KBTS) dataset [44], and CIFAR10 [35].

GTSRB is a well-known benchmark dataset for traffic sign classification. It consists of 43 traffic sign classes. A majority (80%) of the training data (31,367 samples) is divided randomly with a minimum of 900 samples into local client data shards. The remaining 20% of the data (7842 samples) is used for testing.

KUL Belgium traffic sign (KBTS) dataset [44] is another benchmark dataset for traffic sign classification. It consists of 62 traffic sign classes. A majority (80%, i.e., 5562 samples) are used for training with a minimum of 900 samples. The remaining 20% of the data (1416 samples) is considered for testing. We scale the three datasets to an average resolution of 150×150 for our experimentation.

Table 1. CNN configuration

Black-box CNN (4 Conv layers)
input (150 × 150 RGB images)
conv2d_64; kernel 5; stride 1
conv2d_128; kernel 3; stride 1
conv2d_256; kernel 1; stride 1
conv2d_256; kernel 1; stride 1
Fully connected layer 1
Fully connected layer 2
Softmax classifier

CIFAR10 is a well-known benchmark dataset for classification that contains 60,000 samples with ten different classes, namely, airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. A majority (80%) of the training data (48,000 samples) is divided randomly with a minimum of 900 samples into local client data shards. The remaining 20% of the data (12000 samples) is used for testing.

4.2. Base classifier architecture

In this section, we discuss the base classifier architecture used as a global model. We design a customized 4-layer CNN architecture followed by two fully connected layers as shown in Table 1. The model input images with size 150×150 and predict the class probability. The model trained on normal data with three clients resulted in an accuracy of 94.8% and 95.2% for the GTSRB dataset, while 98.3% and 97.0% for the KUL Belgium traffic sign dataset under homogeneous and heterogeneous FL settings, respectively. The proposed CNN network was built with categorical cross-entropy loss function, Adam optimizer. It is trained for 30 and 50 global epochs for GTSRB homogeneous and heterogeneous settings, respectively. For the KUL Beligum traffic sign dataset, the global model is updated for 100 epochs for both FL settings. During the training of the global classifier through FL protocol, each client trains for $E \in [1, 5]$ local epochs on the local data with a batch size $b_s = 64$ and with a learning rate of $l_r = 0.01$.

4.3. Result Discussion

5. Conclusion

References

[1] Jonas Adler and Sebastian Lunz. Banach wasserstein gan. *Advances in neural information processing systems*, 31, 2018. 2

[2] Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. 5

[3] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018. 2

[4] Hana Alghamdi, Mairead Grogan, and Rozenn Dahyot. Patch-based colour transfer with optimal transport. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019. 2

[5] Gil Avraham, Yan Zuo, and Tom Drummond. Parallel optimal transport gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4411–4420, 2019. 2

[6] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020. 4

[7] Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2021. 6

[8] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Model poisoning attacks in federated learning. In *Proc. Workshop Secur. Mach. Learn.(SecML) 32nd Conf. Neural Inf. Process. Syst.(NeurIPS)*, 2018. 4

[9] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *International Conference on Machine Learning*, pages 634–643. ICML, 2019. 1, 2, 3, 4

[10] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017. 3, 6

[11] Alberto Blanco-Justicia, Josep Domingo-Ferrer, Sergio Martínez, David Sánchez, Adrian Flanagan, and Kuan Eeik Tan. Achieving security and privacy in federated learning systems: Survey, research challenges and future directions. *Elsevier Engineering Applications of Artificial Intelligence*, 2021. 1

[12] Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International conference on artificial intelligence and statistics*, pages 880–889. PMLR, 2018. 5

[13] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pages 39–57. IEEE, 2017. 3

[14] Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2

[15] Christian Cosgrove and Alan Yuille. Adversarial examples for edge detection: they exist, and they transfer. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1070–1079, 2020. 2

[16] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30, 2017. 2

Table 2. ..

Method	GTSRB			KBTS			CIFAR10		
	No attack	Single-client attack	Multi-client attack (10/30)	No attack	Single-client attack	Multi-client attack (3/10)	No attack	Single-client attack	Multi-client attack (10/30)
FL	**87.8**	83.24	70.63	**90.02**	83.26	73.14	91.23	85.03	73.83
Random sampling	86.68	84.45	65.45	87.92	84.24	70.53	90.54	82.98	75.33
Power-of-choice	87.56	81.29	63.72	88.05	80.27	69.38	92.64	73.86	70.84
DivFL	87.12	82.63	72.08	89.96	81.63	71.63	92.86	74.12	71.19
FLOT	86.24	85.12	81.12	89.12	**85.94**	**79.94**	91.51	85.21	82.26
Random sampling + FLOT	87.01	**85.98**	**82.26**	89.36	85.02	78.02	**92.37**	**86.24**	**83.54**

Table 3. ..

Method	GTSRB			KBTS			CIFAR10		
	No attack	Single-client attack	Multi-client attack (10/30)	No attack	Single-client attack	Multi-client attack (3/10)	No attack	Single-client attack	Multi-client attack (10/30)
FedAvg	**87.8**	83.24	70.63	**90.02**	83.26	73.14	91.23	85.03	73.83
Krum	86.72	85.80	78.64	89.97	84.29	77.72	91.46	86.12	81.33
Trimmed Mean	84.32	82.87	77.45	88.52	84.09	72.96	90.64	84.43	80.64
Median	85.23	83.39	79.98	88.27	84.97	75.26	89.91	83.36	81.62
FLOT	86.24	85.12	81.12	89.12	**85.94**	**79.94**	91.51	85.21	82.26
Random sampling + FLOT	87.01	**85.98**	**82.26**	89.36	85.02	78.02	**92.37**	**86.24**	**83.54**

[17] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 5

[18] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, et al. Large scale distributed deep networks. *Advances in neural information processing systems*, 25, 2012. 3

[19] Amol Deshpande, Carlos Guestrin, Samuel R Madden, Joseph M Hellerstein, and Wei Hong. Model-based approximate querying in sensor networks. *Springer The VLDB journal*, 14(4):417–443, 2005. 1

[20] Ahmet M Elbir and Sinem Coleri. Federated learning for vehicular networks. *arXiv preprint arXiv:2006.01412*, 2020. 1

[21] Chen Fang, Yuanbo Guo, Yongjin Hu, Bowen Ma, Li Feng, and Anqi Yin. Privacy-preserving and communication-efficient federated learning in internet of things. *Computers & Security*, 103:102199, 2021. 1

[22] Chen Fang, Yuanbo Guo, Na Wang, and Ankang Ju. Highly efficient federated learning with strong privacy preservation in cloud computing. *Computers & Security*, 96:101889, 2020. 1

[23] Farzan Farnia, Amirhossein Reisizadeh, Ramtin Pedarsani, and Ali Jadbabaie. An optimal transport approach to personalized federated learning. *arXiv preprint arXiv:2206.02468*, 2022. 2

[24] R Flamary, N Courty, D Tuia, and A Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Trans. Pattern Anal. Mach. Intell*, 1, 2016. 2, 5

[25] Rémi Flamary, Marco Cuturi, Nicolas Courty, and Alain Rakotomamonjy. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018. 6

[26] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. *Advances in neural information processing systems*, 28, 2015. 6

[27] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2, 3

[28] Xiaojie Guo, Zheli Liu, Jin Li, Jiqiang Gao, Boyu Hou, Changyu Dong, and Thar Baker. V eri fl: Communication-efficient and fast verifiable aggregation for federated learning. *IEEE Transactions on Information Forensics and Security*, 16:1736–1751, 2020. 1

[29] Jamie Hayes and Olga Ohrimenko. Contamination attacks and mitigation in multi-party machine learning. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018. 1

[30] Lie He, Sai Praneeth Karimireddy, and Martin Jaggi. Byzantine-robust learning on heterogeneous datasets via resampling. 2020. 6

[31] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618, 2017. 4

[32] Olakunle Ibitoye, M Omair Shafiq, and Ashraf Matrawy. Differentially private self-normalizing neural networks for adversarial robustness in federated learning. *Computers & Security*, 116:102631, 2022. 1

[33] Leonid V Kantorovich. On the translocation of masses. *Journal of mathematical sciences*, 133(4):1381–1382, 2006. 5

[34] Jakub Konečnỳ, H Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 1

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[36] K. Naveen Kumar, C. Vishnu, Reshmi Mitra, and C. Krishna Mohan. Black-box adversarial attacks in autonomous vehicle

technology. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7, 2020. 2

[37] K Naveen Kumar, C Vishnu, Reshmi Mitra, and C Krishna Mohan. Black-box adversarial attacks in autonomous vehicle technology. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, pages 1–7. IEEE, 2020. 5

[38] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020. 1

[39] Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5764–5772, 2017. 2

[40] Huidong Liu, Xianfeng Gu, and Dimitris Samaras. Wasserstein gan with quadratic transport cost. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4832–4841, 2019. 2

[41] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Zongqi Chen, Xiaoming Huang, and Rongxing Lu. Privacy-enhanced federated learning against poisoning adversaries. *IEEE Transactions on Information Forensics and Security*, 16:4574–4588, 2021. 1

[42] Yanbin Liu, Linchao Zhu, Makoto Yamada, and Yi Yang. Semantic correspondence as an optimal transport problem. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4463–4472, 2020. 2

[43] Samuel R Madden, Michael J Franklin, Joseph M Hellerstein, and Wei Hong. Tinydb: An acquisitional query processing system for sensor networks. *ACM Transactions on database systems (TODS)*, 30(1):122–173, 2005. 1

[44] Markus Mathias, Radu Timofte, Rodrigo Benenson, and Luc Van Gool. Traffic sign recognition — how far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013. 3, 6

[45] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. 1, 3, 6

[46] H. B. McMahan and D. Ramage. Federated learning: Collaborative machine learning without centralized training data (google blogs). 1

[47] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019. 2

[48] Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781. 5

[49] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 2

[50] Viraaji Mothukuri, Reza M Parizi, Seyedamin Pouriyeh, Yan Huang, Ali Dehghantanha, and Gautam Srivastava. A survey on security and privacy of federated learning. *Elsevier Future Generation Computer Systems*, 115:619–640, 2021. 1

[51] Seong Joon Oh, Mario Fritz, and Bernt Schiele. Adversarial image perturbation for privacy protection a game theory perspective. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1491–1500. IEEE, 2017. 2

[52] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017. 2

[53] Shilin Qiu, Qihe Liu, Shijie Zhou, and Chunjiang Wu. Review of artificial intelligence adversarial attack and defense technologies. *Applied Sciences*, 9(5):909, 2019. 2

[54] Julien Rabin, Sira Ferradans, and Nicolas Papadakis. Adaptive color transfer with relaxed optimal transport. In *2014 IEEE international conference on image processing (ICIP)*, pages 4852–4856. IEEE, 2014. 2

[55] Nuria Rodríguez-Barroso, Eugenio Martínez-Cámara, M Luzón, Gerardo González Seco, Miguel Ángel Veganzones, and Francisco Herrera. Dynamic federated learning model for identifying adversarial clients. *arXiv preprint arXiv:2007.15030*, 2020. 1, 2

[56] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000. 2

[57] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, pages 703–718. IEEE, 2022. 3

[58] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving gans using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018. 2

[59] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021. 4

[60] 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)*, pages 1354–1371. IEEE, 2022. 1, 4

[61] Sidak Pal Singh and Martin Jaggi. Model fusion via optimal transport. *Advances in Neural Information Processing Systems*, 33:22045–22055, 2020. 2

[62] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. *arXiv preprint arXiv:1705.10467*, 2017. 1

[63] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *IEEE International Joint Conference on Neural Networks*, pages 1453–1460, 2011. 6

[64] Gan Sun, Yang Cong, Jiahua Dong, Qiang Wang, Lingjuan Lyu, and Ji Liu. Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal*, 2021. 1

[65] Jun Sun, Tianyi Chen, Georgios B Giannakis, Qinmin Yang, and Zaiyue Yang. Lazily aggregated quantized gradient innovation for communication-efficient federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[66] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[67] Sanli Tang, Xiaolin Huang, Mingjian Chen, Chengjin Sun, and Jie Yang. Adversarial attack type i: Cheat classifiers by significant changes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):1100–1109, 2021. 2

[68] Giorgos Tolias, Filip Radenovic, and Ondrej Chum. Targeted mismatch adversarial attack: Query with a flower to retrieve the tower. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5037–5046, 2019. 2

[69] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. Data poisoning attacks against federated learning systems. In *European Symposium on Research in Computer Security*, pages 480–501. Springer, 2020. 4

[70] Luis Caicedo Torres, Luiz Manella Pereira, and M Hadi Amini. A survey on optimal transport for machine learning: Theory and applications. *arXiv preprint arXiv:2106.01963*, 2021. 2

[71] Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the gdpr perspective. *Computers & Security*, 110:102402, 2021. 1

[72] James Tu, Tsunhsuan Wang, Jingkang Wang, Sivabalan Manivasagam, Mengye Ren, and Raquel Urtasun. Adversarial attacks on multi-agent communication. *arXiv preprint arXiv:2101.06560*, 2021. 4

[73] Hongjun Wang, Guanbin Li, Xiaobai Liu, and Liang Lin. A hamiltonian monte carlo method for probabilistic adversarial attack and learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 2

[74] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. *Advances in Neural Information Processing Systems*, 33:16070–16084, 2020. 4

[75] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020. 2

[76] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019. 4

[77] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020. 1

[78] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: distributed backdoor attacks against federated learning. In *8th International Conference on Learning Representations, ICLR*, 2020. 2

[79] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2019. 1

[80] Hongteng Xu, Dixin Luo, Hongyuan Zha, and Lawrence Carin Duke. Gromov-wasserstein learning for graph matching and node embedding. In *International conference on machine learning*, pages 6932–6941. PMLR, 2019. 2

[81] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K. Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020. 2

[82] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pages 5650–5659. PMLR, 2018. 2, 3

[83] Jiale Zhang, Bing Chen, Xiang Cheng, Huynh Thi Thanh Binh, and Shui Yu. Poisongan: Generative poisoning attacks against federated learning in edge computing systems. *IEEE Internet of Things Journal*, 8(5):3310–3322, 2020. 4

[84] Jiale Zhang, Junjun Chen, Di Wu, Bing Chen, and Shui Yu. Poisoning attack in federated learning using generative adversarial nets. In *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 374–380. IEEE, 2019. 1, 2, 4

[85] Lin Zhang, Yong Luo, Yan Bai, Bo Du, and Ling-Yu Duan. Federated learning for non-iid data via unified feature learning and optimization objective alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4420–4428, October 2021. 3