Enhancing Federated Learning Robustness using Locally Benignity-Assessable Bayesian Dropout

Jingjing Xue, Sheng Sun, Min Liu *Senior Member, IEEE*, Qi Li *Senior Member, IEEE*, and Ke Xu, *Fellow, IEEE*

***Abstract*—Federated Learning (FL) has emerged as a privacy-preserving training paradigm, which enables distributed devices to jointly learn a shared model without raw data sharing. However, the inaccessible client-side data and unverifiable local training leave FL vulnerable to Byzantine attacks. Most defense strategies focus on penalizing malicious clients in server-side aggregations and ignore clients-side weight units poisoning assessment, failing to maintain robustness and convergence in non-IID settings. In this paper, we propose Federated learning with Benignity-assessable Bayesian Dropout and variational Attention (FedBDA) to achieve local robust training based on fine-grained benignity indicators and guarantee global robustness over non-IID data. Specifically, FedBDA integrates variational inference explanation of dropout into local training, where each client individually quantifies the benign degree of weight units to determine a resilient dropping pattern for the local Bayesian model, enabling client-side robust training with Bayesian interpretability. To accommodate variational distributions of local Bayesian models and globally assess their benign potentials, we design a joint attention mechanism based on Jensen-Shannon divergence among local, global, and median distributions for robust weighted aggregation. Theoretical analysis proves the robustness and convergence of FedBDA. We conduct extensive experiments on four benchmark datasets with five typical attacks, and the results demonstrate that FedBDA outperforms status quo approaches in model performance and running efficiency.**

***Index Terms*—Federated learning, Byzantine attack, dropout defense, robust aggregation**

I. Introduction

The growing concerns over data privacy have triggered significant interest in Federated Learning (FL), a distributed training paradigm [1], [2] that enables edge devices to collaboratively learn a global model without raw data exchange. In particular, resource-constrained edge devices (*e.g.*, smartphones) locally maintain their data and individually train local models to get client-side updates, which are periodically uploaded to a central server. Subsequently, the server aggregates received local updates to obtain a global model.

Jingjing Xue is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and also with the University of Chinese Academy of Sciences, Beijing, China. (e-mail: xuejingjing20g@ict.ac.cn)

Min Liu is with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and also with the Zhongguancun Laboratory, Beijing, China. (e-mail: liumin@ict.ac.cn).

Sheng Sun is with Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China (e-mail: sunsheng@ict.ac.cn).

Qi Li is with the Institute for Network Sciences and Cyberspace and Beijing National Research Centre for Information Science and Technology (BNRist), Tsinghua University, Beijing, China. China (e-mail: qli01@tsinghua.edu.cn).

Ke Xu is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, and also with the Zhongguancun Laboratory, Beijing, China. (e-mail: xuke@tsinghua.edu.cn.)

Corresponding author: Min Liu.

Despite the benefits in data location and privacy protection, FL remains vulnerable to Byzantine attacks from distributed devices (*i.e.*, clients) due to the inaccessible local data and training procedure [3], [4]. Byzantine attackers can stealthily invade clients and manipulate local training [5], [6]. The clients compromised by attackers (*i.e.*, malicious clients) inject fake parameters into local models or modify data labels, bringing wrong local updates. With the global aggregation, malicious updates further contaminate the global model, thereby deteriorating accuracy and hampering FL convergence [7].

Against Byzantine attacks, previous defense methods focus on global careful aggregations [8], [9], [10], where the server penalizes likely malicious updates and aggregates other updates in resilient rules [11]. Some statistical properties like median are utilized to circumvent poisoning outliers [12]. Distance-based defenses such as Krum [13] and Clipped-Cluster [10] examine the distance or similarity betweenlocal updates to identify malicious clients and weaken or exclude them from aggregation. However, these methods rely on a strong assumption of Independent and Identically Distributed (IID) data [3], which are ineffective in practical scenarios with non-IID data [14], [15]. Elaborate attack designs in pioneering studies [16], [17] can bypass extant resilient aggregations [18].

In actual FL settings, heterogeneous data and different poisoning states have superimposed effects on local updates such that the attack situation of clients cannot be accurately reflected by the global-view evaluation of pairwise entire local models. Non-IID data across clients leads to inconsistent local updates regardless of whether it is attacked [19], [20]. Besides, Byzantine attacks can generate distinct malicious effects on different weight units (*e.g.*, recurrent rows or channel matrices) of local models [21]. Therefore, the fine-grained benign assessment of weight units is required from the local view, so malicious error can be promptly eliminated to achieve local robust training and cooperate with global resilient aggregation.

To this end, we propose Federated learning with Benignity-assessable Bayesian Dropout and variational Attention (FedBDA), a novel Byzantine-robust framework that enables local and global benignity assessments at different granularities. Specifically, FedBDA involves two core modules: (i) Bayesian adaptive dropout by locally quantifying weight unit-wise benign effect and (ii) variational attention-based aggregation with globally JS divergence-dependent maliciousness evaluation of entire model distributions. First, we integrate variational Bayesian inference with local dropout training and allow each client to adaptively quantify the benign score indicator of weight units over client-side data. Guided by local

benign indicators, clients can promptly drop out poisonous weight units and customize robust dropping patterns for variational distributions of client-side Bayesian models, thereby mitigating the malicious impact of wrong information during local training. Then, only non-zero variational parameters after dropout are uploaded to the server, which is beneficial for reducing uplink costs. Second, on the server side, variational distributions of different clients still have distinct impacts on global generalization and robustness. A variational attention mechanism is designed to globally evaluate the benign potentials of local variational distributions according to joint metrics of Jensen-Shannon (JS) divergences among local, global, and median distributions, further achieving server-side resilient aggregation. In this way, FedBDA provides dual local-global robustness guarantees in non-IID settings.

Contributions. Our contributions are summarized as follows:

- We propose a robust and efficient FL framework against Byzantine attacks, termed FedBDA, which takes the first step to quantify the benign contribution of each weight unit on client-specific data in a Bayesian interpretable way, improving Byzantine robustness in non-IID settings with less uplink communication overhead.
- We design a benignity-guided Bayesian dropout to customize resilient dropping patterns for local variational distributions, enabling client-side robust training. To accommodate local updates of variational distributions and realize resilient aggregation, a variational attention mechanism is developed for server-side robustness guarantee.
- We theoretically prove the robustness of FedBDA and further analyze the generalization error bound and convergence property of FedBDA on heterogeneous data.
- We conduct extensive experiments to evaluate FedBDA against various Byzantine attacks on real-world datasets. The results show that FedBDA provides 1.14%-11.74% benign accuracy gains on highly non-IID data with up to $2\times$ uplink costs reduction, which brings up to 56.9% running time saving compared to state-of-the-art baselines.

II. Related Work

Byzantine Attacks. The Byzantine attack is a typical security threat in FL, which can intrude and control clients to violate federation protocols and transmit malicious updates to the server [22], thereby injecting wrong information into the global model [11]. Current Byzantine attacks can be roughly classified into two types: the training data-based attack and the model parameter-based attack. The former tampers with training data labels and further injects incorrect knowledge into local models during training [23], [24]. For example, Label Flipping (LF) attack [25] flips the labels of a chosen data subset. The latter directly crafts poisoning parameters and uploads them to the server, [9], [26]. Typical parameter-based attacks include "A Little is Enough" (ALIE) [16], Inner Product Manipulation (IPM) [27], and Sign Flipping (SF).

Byzantine-Robust FL. Many resilient aggregations have been proposed to defend against Byzantine attacks [9], [10], [28], which can be roughly categorized into three directions. *(1) Performance-driven decisions* [29] test the performance of local models in the server to detect malicious clients, which additionally require a central clean dataset for performance verification. However, the auxiliary data still involves the privacy problem, hindering their practicability [9]. *(2) Statistical aggregations* calculate statistics measures of local updates as new global parameters. For instance, coordinate-wise median and trimmed mean of local parameters are regarded as representatives of a majority of clients in [12]. These statistics are more likely to be skewed towards poisoning updates as malicious clients increase. *(3) Distance-based defenses* try to distinguish poisoned updates according to distance or similarity differences. For example, FoolsGold [14] exploits inter-client cosine similarity to detect abnormal updates. These defenses are only feasible when a malicious update is dispersed and deviates from all benign updates.

Furthermore, several pioneering studies explore auxiliary defense techniques to collaborate with resilient aggregations. GAS [28] splits high-dimensional local updates into sub-vectors in the server and then applies robust aggregation to sub-vectors in non-IID settings, which still only focuses on detecting malicious updates from the global view and ignores client-side robust optimization. Dropout [30] is also regarded as an effective auxiliary defense. FedREP [31] and FLAP [32] integrate dropout with robust aggregation to mitigate attack effects while improving communication efficiency. FLAP prunes the global model after the aggregation, which does not involve client-side robust optimization and convergence support. FedREP directly zeros out lower-magnitude weights without considering the malicious effect of model weights on local performance. Both cannot adapt to non-IID data across clients in real FL scenarios, leading to robustness degradation and convergence disorder. Unlike the aforementioned methods, this work locally assesses benign score of each weight unit and independently drops out poisoning and insignificant weight units of client-side Bayesian models during local training to promptly mitigate malicious errors and achieve client-side robust training in non-IID settings.

III. Problem Statement

A. FL with Bayesian Neural Networks

We consider a FL system with K devices (*i.e.*, clients) and a central server. Each client $k \in \mathcal{K}$ possesses its data $\mathcal{D}^k$ and a true function f_0^k, where $y^k = f_0^k(\boldsymbol{x}^k)$ for $(\boldsymbol{x}^k, y^k) \in \mathcal{D}^k$. The data $\mathcal{D}^k$ are utilized to train the DNN model to estimate f_0^k. Considering the generalization and theoretical interpretability, Bayesian Neural Network (BNN) is introduced into FL [33], which offers a probabilistic interpretation and a measure of uncertainty for DNN [34], [35]. In BNN, each model weight is viewed as a random variable denoted by Gaussian distribution, called the prior distribution. By incorporating knowledge of prior distributions of model weights and a functional form of the likelihood, we infer the posterior distribution of model weights. For modern Bayesian models, exact computation of the posterior is intractable due to the high dimensionality and non-convexity, so approximate posterior inference is resorted. Variational inference is a popular solution for approximate

posterior [36]. In this work, we focus on variational Bayesian inference in FL. Specific expressions of variational Bayesian inference are elaborated in Section IV.

By applying variational Bayesian inference in FL, each client k locally learns a BNN model represented by an approximate posterior distribution $\tilde{\pi}^k(\theta|\mathcal{D}^k) \in \mathcal{F}$ [37], [38], [39]. Here, θ is a model weight set, and $\mathcal{F}$ denotes a feasible model distribution set. The local model distribution $\tilde{\pi}^k(\theta|\mathcal{D}^k)$ is determined by the posterior mean and variance parameters, as referred to [33]. We define the variational parameter set U^k to represent the posterior mean of the model distribution. Following [35], [40], we suppose that the posterior variance $\tilde{s}^2$ is constant for all clients and is not updated and transmitted. The local updates of $\tilde{\pi}^k(\theta|\mathcal{D}^k)$ only involve the variational parameters, which are periodically uploaded to the server. Then, the server can get a global Bayesian model denoted by the global distribution $\tilde{\pi}^g \sim \mathcal{N}(U, \tilde{s}^2 I)$, where U denotes the global variational parameters obtained by aggregating all local updates. In practical applications, the distributed dataset $\{\mathcal{D}^1, \ldots, \mathcal{D}^k\}$ is essentially non-IID, and the uplink is typically much slower than the downlink (*e.g.*, 17.6Mbps up vs. 204.9Mbps down in the T-Mobile 5G network [41]). We explore a Bayesian inference-based resilient dropout for clients in such non-IID and limited uplink bandwidth scenarios.

B. Byzantine Attack Modeling in FL

In FL, Byzantine attackers manipulate clients, but do not compromise the server [29]. The goal of the adversary is to degrade global performance and impair convergence. Attackers have the capability to know the local training data of malicious clients and local updates of all clients. We suppose that malicious clients still complete local training and parameter upload of the current round with poisoning data or interfered parameters after being attacked. The server can only access global parameters and local updates without other knowledge.

Let $\mathcal{A}$ denote the malicious client set, which is fixed over time. Aligned with prior works [7], [18], we assume that the number of malicious clients holds $|\mathcal{A}| < K/2$. The rest clients $\mathcal{K}\backslash\mathcal{A}$ are benign and will faithfully implement local training. If attackers modify partial data labels of client $k \in \mathcal{A}$, we define a local poisonous dataset as $\mathcal{D}^k_{\mathcal{A}}$, and other data $\mathcal{D}^k\backslash\mathcal{D}^k_{\mathcal{A}}$ are honest. In this setting, we aim at learning the optimal global Bayesian model to minimize the average loss over all honest data across clients, formulated as:

$$\min_{\tilde{\pi}^g \in \mathcal{F}} \frac{1}{K} \Bigg\{ \sum_{k \in \mathcal{K}\backslash\mathcal{A}} \frac{\mathcal{L}^k(\tilde{\pi}^g; \mathcal{D}^k)}{|\mathcal{D}^k|} + \sum_{k \in \mathcal{A}} \frac{\mathcal{L}^k(\tilde{\pi}^g; \mathcal{D}^k\backslash\mathcal{D}^k_{\mathcal{A}})}{|\mathcal{D}^k\backslash\mathcal{D}^k_{\mathcal{A}}|} \Bigg\}. \quad (1)$$

Here, $\mathcal{L}^k(\cdot; \mathcal{D}^k)$ denotes the local loss over $\mathcal{D}^k$.

IV. Preliminary

A. Variational Bayesian Inference of DNN

A Deep Neural Network (DNN) f_θ learned from the dataset (X, Y) is used to approximate an unknown true function f_0 such that $\boldsymbol{y} = f_0(\boldsymbol{x})$ and $f_\theta(\boldsymbol{x}) \approx f_0(\boldsymbol{x})$ for any $(\boldsymbol{x}, \boldsymbol{y}) \in (X, Y)$, where θ is the weight set. Two common types of DNN, *i.e.*, non-recurrent neural network and Recurrent Neural Network (RNN) are considered. We recursively express a non-recurrent neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^o$ with L layers as:

$$\boldsymbol{x}_l = \phi(W_l \boldsymbol{x}_{l-1}), \;\; f_\theta(\boldsymbol{x}) = \boldsymbol{x}_L \quad \text{for } \; l = 1, \ldots, L,$$

where d is the input dimension and o denotes the output dimension. ϕ denotes an activation function and $\boldsymbol{x}_0 = \boldsymbol{x}$. Besides, W_l denotes the weight matrix in the l-th layer such that $\theta = \{W_1, \ldots, W_L\}$. As referred to [35] and [33], we employ an equal-width DNN to facilitate analysis, meaning that the hidden dimension of all layers is equal to D, where $d \leq D$ and $o \leq D$. For sequence tasks, the RNN is more powerful. In RNN models, the input is an embedding sequence $\boldsymbol{x} = [x_1, \ldots, x_L]^\top \in \mathbb{R}^{L \times d}, x_l \in \mathbb{R}^d$, which is mapped by

$$h_l = \varrho(W_x x_l + W_h h_{l-1}), \; f_\theta(\boldsymbol{x}) = \rho(h_L) \;\text{ for } l = 1, \ldots, L.$$

ϱ and ρ are activation functions, and L deno dtes a fixed input sequence length. The input-hidden weight matrix $W_x \in \mathbb{R}^{D \times d}$ and hidden-hidden weight matrix $W_h \in \mathbb{R}^{D \times D}$ (*i.e.*, recurrent connections of RNN) consist of the weight set $\theta = \{W_x, W_h\}$.

With Bayesian inference theory [40], [35], θ can be regarded as random variables denoted by prior π, and the target of model training is to calculate the posterior $\pi(\theta|X, Y)$. For easier posterior estimation, we introduce the tempered posterior $\pi_{m,\alpha} \propto \sum_{i=1}^m \alpha \ln p(\boldsymbol{y}_i|\boldsymbol{x}_i, \theta)\pi(\mathrm{d}\theta)$, mentioned in [42]. Here, $\alpha \in (0, 1)$, and m denotes the number of training samples. With Variational Inference (VI) [37], [43], a variational approximation $\tilde{\pi}$ is learned to estimate the tempered posterior $\pi_{m,\alpha}$. First, we define

Definition IV.1. The generalization error of the variational approximation $\tilde{\pi}$ is $G_e(\tilde{\pi}) = \mathbb{E}\Big[\int \|f_\theta - f_0\|_2^2 \tilde{\pi}(\mathrm{d}\theta)\Big]$.

For the optimization of variational approximation $\tilde{\pi}$, the loss function is represented as

$$\mathcal{L}(\tilde{\pi}; \mathcal{D}) = \frac{\alpha}{2\sigma^2} \sum_{i=1}^m \int \big(\boldsymbol{y}_i - f_\theta(\boldsymbol{x}_i)\big)^2 \tilde{\pi}(\mathrm{d}\theta) + KL(\tilde{\pi}\|\pi), \quad (2)$$

where σ^2 is the likelihood variance, *i.e.*, $p(\boldsymbol{y}_i|\boldsymbol{x}_i, \theta)$ follows a Gaussian distribution with the variance σ^2 for any $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{D}$ with $\mathcal{D} = (X, Y)$. The second term is approximated to L2 regularization [44], beneficial to avoid overfitting. The local loss $\mathcal{L}^k(\cdot; \mathcal{D}^k)$ in optimization target (1) is calculated by (2).

B. Bayesian Interpretation of Dropout

Defining S as the number of nonzero parameters in the local model, the sparse model is constructed by (S, L, D), where $\Theta_{S,L,D}$ denotes the feasible weight space. We focus on structured dropout, which involves a structure-level dropping granularity. The model weights of the smallest network unit that can be represented by the dropping granularity are defined as the weight unit. We consider that a DNN model contains J sparsifiable weight units. The dropout zeros out some weight units, which can be accomplished by placing spike-and-slab distributions [35] over the DNN model. Inspired by [40], the distribution of the j-th weight unit $\mathbf{w}_j$ can be expressed as

$$\tilde{\pi}_{\mathbf{w}_j} = (1 - p)\mathcal{N}(\boldsymbol{\mu}_j, \tilde{s}^2 I) + p\delta(0), \quad (3)$$

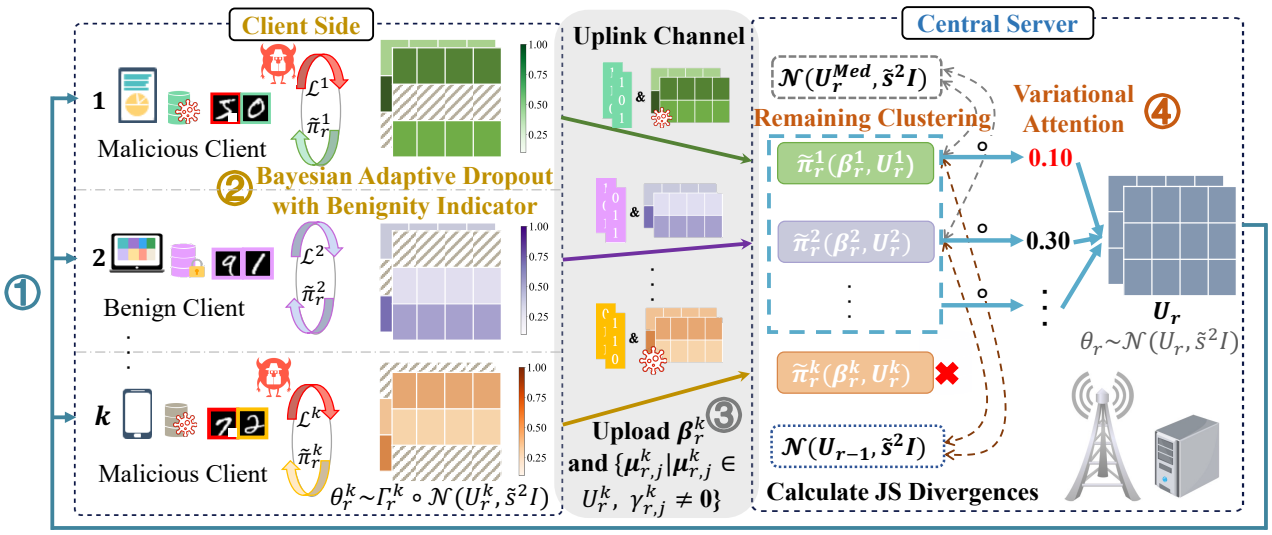


Fig. 1: The overview diagram of FedBDA. In each round, ① the server randomly selects clients and sends global variational parameters to them. ② Each selected client locally trains model distributions with Bayesian adaptive dropout according to the unit-wise benignity indicator. ③ After training, the non-zero variational parameters and binary dropping pattern are uploaded to the server. ④ The server robustly aggregates local variational parameters with JS divergence-based attention scores.

where $\boldsymbol{\mu}_j$ denotes variational parameters that need to be optimized in local training, and $\tilde{s}^2$ is the posterior variance. Besides, p is a dropout rate, and $\delta(0)$ denotes an impulse function. For the clear representation of the dropout, we introduce the dropping pattern $\Gamma = [\gamma^1, \ldots, \gamma^J]^\top \in \{0,1\}^J$ to point out which weight units are masked. If $\gamma_j = 0$, the j-th weight unit is zeroed out.

V. The FedBDA Framework

A. Overview of FedBDA

In this study, we propose the FedBDA framework to achieve local robust training with Bayesian interpretability and enable dual local-global robustness in non-IID settings. Considering the superimposed effects of non-IID data and poisoning states on local models, FedBDA assesses the contributions of weight units to the performance gain from a local view and quantifies them as unit-wise benignity indicators. The benignity indicator induces a resilient dropping pattern for the variational distribution of the local Bayesian model, which promptly mitigates the malicious impact of poisoned weight units, facilitating local robust training. To accommodate local updates of variational distribution, we design variational attention based on JS divergence for the global aggregation, guaranteeing the generalization and robustness of the global model. Figure 1 illustrates the procedures of FedBDA in a round, which involves: ① The server randomly selects clients to participate in the training of the current round and sends global variational parameters to them. ② Each selected client locally trains model distributions with Bayesian adaptive dropout based on the benignity assessment of weight units. ③ After training, the non-zero variational parameters and binary dropping pattern are uploaded to the server. ④ Finally, the server aggregates local variational parameters with variational attention to obtain the global model distribution. Specifically, the two core designs of FedBDA are as follows:

- **Local training with Bayesian adaptive dropout:** Each client initializes the local model with receiving global variational parameters and then drops out partial weight units. During local training, the client adaptively adjusts dropping patterns based on the changing trend of training loss. The dropout experiences are accumulated in a score vector to reflect the benign effect of weight units. After training, the nonzero variational parameters and binary dropping pattern (much less than variational parameters and even can be ignored) are uploaded to the server.
- **Global aggregation with variational attention:** The server reconstructs client-side variational distributions and filters deemed malicious clients by clustering. Then, by measuring JS divergences among local, global, and median approximations, different attentions are assigned to local variational parameters for robust weighted aggregation. Iteratively, the server randomly selects clients for training in the next round and sends newly aggregated variational parameters to them.

Detailed process is summarized in Algorithm 1.

B. Bayesian Adaptive Dropout with Benignity Indicator

In the whole FL process, Byzantine attackers continuously inject poisonous information into malicious clients, while benign clients are contaminated by the delivered poisoning global model. Regardless of attacked or benign clients, malicious errors always remain during local training, and different weight units exhibit diverse malicious impacts on each client [45]. Superimposing non-IID data, the same unit has distinct impact on different clients. Therefore, it is crucial to granularly distinguish the effect of weights unit from the local view to promptly alleviate malicious impact for local robust training.

Algorithm 1: FedBDA

Input: dropout rate p; client selection fraction κ; local datasets $\{\mathcal{D}^1, \ldots, \mathcal{D}^K\}$; global training round R; the number of local iterations V; stage boundary τ_R.
Initialize: global variational parameters U_0; the number of selected clients $c \leftarrow \max(\lfloor \kappa \cdot K \rfloor, 1)$; dropping pattern set $\mathcal{Z}_S$; posterior variance $\tilde{s}^2$.
Output: global variational approximation $\tilde{\pi}^g = \mathcal{N}(U_R, \tilde{s}^2 I)$
ServerRun:
1: **for** each round $r = 1, 2, \ldots, R$ **do**
2: $\mathcal{C}_r \leftarrow$ randomly select c clients
3: Transmit U_{r-1} to each selected client
4: **for** each client $k \in \mathcal{C}_r$ **in parallel do**
5: $\Gamma_r^{k,V}$, $\{\boldsymbol{\mu}_{r,j}^{k,V}\} \leftarrow$ **LocalUpdate**(U_{r-1}, r)
6: **end for**
7: Reconstruct local spike-and-slab distribution $\tilde{\pi}_r^{k,V} = \Gamma_r^{k,V} \mathcal{N}(\Gamma_r^{k,V} \circ U_r^{k,V}, \tilde{s}^2 I) + (1 - \Gamma_r^{k,V})\delta(0)$
8: $U_r \leftarrow$ **VarAttAgg**$(\{\tilde{\pi}_r^{k,V} | k \in \mathcal{C}_r\}, U_{r-1})$
// variational attention-based aggregation by Algorithm 2
9: **end for**
10: **return** $\tilde{\pi}^g = \mathcal{N}(U_R, \tilde{s}^2 I)$

LocalUpdate(U_{r-1}, r)**:**
11: Set $U_r^{k,0} \leftarrow U_{r-1}$ and $\theta_r^{k,0} \sim \mathcal{N}(U_r^{k,0}, \tilde{s}^2 I)$
12: **if** $r \leq \tau_R$ **then**
13: $\Gamma_r^{k,0} = \ldots = \Gamma_r^{k,2\tau} \leftarrow$ Draw from $\mathcal{Z}_S$ randomly *// stage one*
14: **else**
15: $\Gamma_r^{k,0} = \ldots = \Gamma_r^{k,V} \leftarrow$ Calculate based on local benignity indicator B^k *// stage two*
16: **end if**
17: **for** each iteration $v = 0, 1, \ldots, V-1$ **do**
18: $\theta_r^{k,v} \sim \tilde{\pi}_r^{k,v} = \Gamma_r^{k,v} \circ \mathcal{N}(U_r^{k,v}, \tilde{s}^2 I)$
19: $U_r^{k,v+1} \leftarrow U_r^{k,v} - \eta\big[\Gamma_r^{k,v} \circ \nabla_U \mathcal{L}^k(\tilde{\pi}_r^{k,v}; \mathbf{d}_v^k)\big]$, $\mathbf{d}_v^k \in \mathcal{D}^k$
20: **if** $r \leq \tau_R$ **and** $v > \tau$ **and** $v \,\%\, \tau = 0$ **then**
21: Calculate $\Delta\mathcal{L}_r^{k,v}$ using (4)
22: **if** $\Delta\mathcal{L}_r^{k,v} > 0$ **then**
23: $\Gamma_r^{k,v+1} = \ldots = \Gamma_r^{k,v+\tau} \leftarrow$ Draw from $\mathcal{Z}_S$ randomly
24: **else**
25: $\Gamma_r^{k,v+1} = \ldots = \Gamma_r^{k,v+\tau} \leftarrow \Gamma_r^{k,v}$
26: **end if**
27: Update benignity indicator B^k by (5)
28: **end if** *// only execute in stage one*
29: **end for**
30: **return** $\Gamma_r^{k,V}$, $\{\boldsymbol{\mu}_{r,j}^{k,V} | \boldsymbol{\mu}_{r,j}^{k,V} \in U_r^{k,V}, \gamma_{r,j}^{k,V} \neq 0\}$

Unit-wise Benignity Indicator. To locally quantify the effect of weight units, we develop a benignity indicator for each client, which accumulates experience in local performance gains of weight unit dropout. With the benignity indicator, the client customizes robust dropping patterns for variational distributions to adaptively drop out malicious and insignificant weight units, enabling client-side robust training.

For local performance gains of weight units, we consider the ability of each unit to facilitate loss reduction. Specifically, a fixed iteration interval τ is preset to evaluate change trends of training loss, and client k maintains the same dropping pattern in such τ iterations. Suppose $\mathbf{d}_v^k$ is a batch of training samples in the v-th iteration, we define $\hat{\mathcal{L}}_r^{k,v} = \sum_{i=v-\tau+1}^{v} \mathcal{L}^k(\tilde{\pi}_r^{k,i}; \mathbf{d}_i^k)$ as the average loss of the $(v-\tau+1)$-th iteration to the v-th iteration in round r. The loss gap between adjacent τ iterations

$$\Delta\mathcal{L}_r^{k,v} = \hat{\mathcal{L}}_r^{k,v} - \hat{\mathcal{L}}_r^{k,v-\tau}, \quad v \geq 2\tau \tag{4}$$

can represent the loss change caused by current dropping pattern $\Gamma_r^{k,v}$ of client k in round r. $\Delta\mathcal{L}_r^{k,v} > 0$ indicates that the current dropping pattern effectively facilitates loss decrease and defends against attacks, which can be kept in the next τ iterations. Otherwise, client k adjusts local dropping patterns by resampling from the feasible pattern set $\mathcal{Z}_S$. Then, we update the local benignity indicator $B^k = [b_1^k, \ldots, b_J^k]^\top \in \mathbb{R}^J$. If the j-th weight unit is held in the v-th iteration, the benignity score b_j^k is updated in the next iteration by

$$b_j^k = \begin{cases} b_j^k + 1, & \text{if } \Delta\mathcal{L}_r^{k,v} \leq 0, \\ b_j^k + \gamma_j^{k,v+1}, & \text{if } \Delta\mathcal{L}_r^{k,v} > 0. \end{cases} \tag{5}$$

A higher score means that the unit is more conducive to performance gain and should be retained. The benignity indicator is fed back to determine a resilient dropping pattern such that poisonous units are promptly pruned for local robust training.

Local Update. The function **ClientUpdate**$(\cdot,\cdot)$ in Algorithm 1 describes the local update process. Specifically, given dropout rate p, we have non-zero parameter number $S = (1-p)\mathrm{J} \times D$. The feasible pattern set $\mathcal{Z}_S$ is also specific. In round r, the selected client $k \in \mathcal{C}_r$ initializes variational parameter $U_r^{k,0} \leftarrow U_{r-1}$ and samples the local model $\theta_r^{k,0} \sim \mathcal{N}(U_r^{k,0}, \tilde{s}^2 I)$ (line 11). Then, a dropping pattern is required to remove malicious weight units in the local model. For pattern determination, Bayesian adaptive dropout can be divided into two stages by the round boundary τ_R. In *stage one* (*i.e.*, global round $r \leq \tau_R$), client $k \in \mathcal{C}_r$ samples the initial dropping pattern $\Gamma_r^{k,0}$ from $\mathcal{Z}_S$ (line 13). Each weight unit with the dropping label $\gamma_{r,j}^{k,0} = 0$ is zeroed out such that the initial variational distribution of client k is denoted as

$$\tilde{\pi}_r^{k,0}(\Gamma_r^{k,0}, U_r^{k,0}) = \Gamma_r^{k,0} \mathcal{N}(U_r^{k,0}, \tilde{s}^2 I) + (1 - \Gamma_r^{k,0})\delta(0) \tag{6}$$

for sparse model characterization. Subsequently, this sparse model is locally trained. For the v-th iteration, the variational parameters $U_r^{k,v} = [\boldsymbol{\mu}_{r,1}^{k,v}, \ldots, \boldsymbol{\mu}_{r,\mathrm{J}}^{k,v}]^\top$ of local spike-and-slab distribution $\tilde{\pi}_r^{k,v}$ are updated by

$$U_r^{k,v+1} = U_r^{k,v} - \eta\big[\Gamma_r^{k,v} \circ \nabla_U \mathcal{L}^k\big(\tilde{\pi}_r^{k,v}(\Gamma_r^{k,v}, U_r^{k,v}); \mathbf{d}_v^k\big)\big] \tag{7}$$

with the learning rate η. At intervals of τ iterations, client k calculates the training loss gap by (4) and adaptively adjusts dropping patterns, as reported in lines 21-26. Moreover, the dropout experiences of weight units are accumulated in benignity indicator B^k via (5) at each pattern adjustment.

Until $r > \tau_R$, FedBDA enters *stage two*, where the benignity indicator B^k is adequate to customize a high-quality dropping pattern (line 15). The weight units with lower benign scores exhibit weak performance gains over local data in historical exploration, which are discarded preferentially. With the dropout rate p, client k computes a score threshold λ^k as the p-quantile of B^k such that the benignity indicator-based dropping pattern of client k in round $r > \tau_R$ is formulated by

$$\Gamma_r^{k,v} = \psi(B^k - \lambda^k I), \quad \text{for } v \in \{0, \ldots, V-1\}, \tag{8}$$

where $\psi(\cdot)$ is a step function. The remaining variational parameters $\Gamma_r^{k,v} \circ U_r^{k,v}$ are locally trained by (7) for client-side robust optimization. After V local iterations, only nonzero variational parameters $\{\boldsymbol{\mu}_{r,j}^{k,V} | \boldsymbol{\mu}_{r,j}^{k,V} \in U_r^{k,V}, \gamma_{r,j}^{k,V} \neq 0\}$ and

binary dropping pattern $\Gamma_r^{k,V}$ are transmitted to the server, which mitigates uplink communication overhead.

C. Variational Attention-based Aggregation

The server can reconstruct variational distributions of local Bayesian models based on received variational parameters and dropping patterns. The next step is to aggregate them into a robust global approximation. Traditional average aggregation cannot penalize malicious updates such that the global model deviates from what it should be optimized [29], further affecting local updates of benign clients. As shown in Figures 2, the LF attack significantly changes the distributions of local model updates even for benign clients under the classical FedAvg framework [1]. Moreover, existing robust aggregations [10], [14], [28] focus on weight values with certainty and fail to analyze model distributions with Bayesian uncertainty. Thus, there is a need for global maliciousness evaluation of local variational distributions with Bayesian uncertainty to guarantee robust aggregation of variational distributions.

Clustering for Local Variational Distributions. In view of the significant derivations of some malicious updates, clustering has shown a superior detection capability for poisoning updates. We introduce a JS divergence-dependent clustering to accommodate local variational distributions with Bayesian uncertainty. Specifically, the server first reconstructs local variational distributions $\{\tilde{\pi}_r^{k,V} | k \in \mathcal{C}_r\}$ by

$$\tilde{\pi}_r^{k,V} = \left[\tilde{\pi}_{\mathbf{w}_{r,1}^{k,V}}, \ldots, \tilde{\pi}_{\mathbf{w}_{r,j}^{k,V}}, \ldots, \tilde{\pi}_{\mathbf{w}_{r,J}^{k,V}}\right]^\top \tag{9}$$

with

$$\tilde{\pi}_{\mathbf{w}_{r,j}^{k,V}} = \begin{cases} \mathcal{N}(\boldsymbol{\mu}_{r,j}^{k,V}, \tilde{s}^2 I), & \text{if } \gamma_{r,j}^{k,V} \neq 0, \\ \delta(0), & \text{if } \gamma_{r,j}^{k,V} = 0. \end{cases} \tag{10}$$

Here, $\mathbf{w}_{r,j}^{k,V}$ denotes the j-th weight unit of the local model uploaded by client k. Thus, we can rebuild local variational distributions with the form of (6). The JS divergence between pairwise variational distributions for clients $k_1, k_2 \in \mathcal{C}_r$ is

$$\begin{aligned} \chi_r^{k_1,k_2} &= JS(\tilde{\pi}_r^{k_1,V} \| \tilde{\pi}_r^{k_2,V}) \\ &= \frac{KL\big(\tilde{\pi}_r^{k_1,V} \| (\tilde{\pi}_r^{k_1,V} + \tilde{\pi}_r^{k_2,V})/2\big)}{2} \\ &\quad + \frac{KL\big(\tilde{\pi}_r^{k_2,V} \| (\tilde{\pi}_r^{k_1,V} + \tilde{\pi}_r^{k_2,V})/2\big)}{2}. \end{aligned} \tag{11}$$

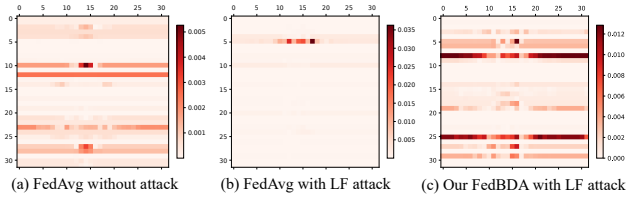
Based on JS divergences among all selected clients, we separate clients into two clusters $\mathcal{C}_r^1, \mathcal{C}_r^2$, where

$$\mathcal{C}_r^1, \mathcal{C}_r^2 = \arg \min_{\mathcal{C}_r^1 \cup \mathcal{C}_r^2 = \mathcal{C}_r} \Big(\max_{k_1 \in \mathcal{C}_r^1, k_2 \in \mathcal{C}_r^2} \chi_r^{k_1,k_2} \Big), \tag{12}$$

as described in lines 1-6 of Algorithm 2. Following [46], we suppose that most clients in $\mathcal{C}_r$ are benign and belong to a larger cluster, while other clients are deemed malicious and should be excluded from global robust aggregation. Defining $\mathcal{C}_r^p$ as the remaining client set after clustering, it is denoted by

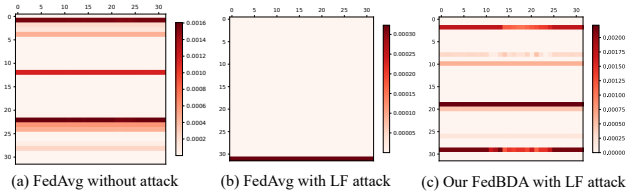
$$\mathcal{C}_r^p \leftarrow \arg \max_{\mathcal{C} \in \{\mathcal{C}_r^1, \mathcal{C}_r^2\}} |\mathcal{C}|, \tag{13}$$

and only local variational parameters of client $k \in \mathcal{C}_r^p$ participate in subsequent aggregation.



(a) FedAvg without attack (b) FedAvg with LF attack (c) Our FedBDA with LF attack

(a) Partial parameter distribution of local model updates on the benign client.



(a) FedAvg without attack (b) FedAvg with LF attack (c) Our FedBDA with LF attack

(b) Local partial update distribution of the malicious client in round R.

Fig. 2: The comparison of partial parameter distributions of local model updates with or without attack.

Variational Attention. Among the remaining local updates after clustering, there may still be residual maliciousness for the global model in non-IID settings. Hence, we design a joint metric to measure benign potentials for global generalization and robustness based on JS divergences among local, global, and median distributions. The global variational approximation is denoted as $\tilde{\pi}_{r-1}^g = \mathcal{N}(U_{r-1}, \tilde{s}^2 I)$. The difference between local variational distributions and the global approximation reflects the generalization [47], which is expressed as

$$\chi_r^{k,g} = JS(\tilde{\pi}_r^{k,V} \| \tilde{\pi}_{r-1}^g), \quad \forall\, k \in \mathcal{C}_r^p. \tag{14}$$

Considering the longstanding history of median in robust statistics [48], we introduce U_r^{Med} as the median of the remaining variational parameters. The median approximation is denoted as $\tilde{\pi}_r^{Med} = \mathcal{N}(U_r^{Med}, \tilde{s}^2 I)$. The JS divergence between local variational distributions and the median approximation is associated with global robustness, denoted as

$$\chi_r^{k,Med} = JS(\tilde{\pi}_r^{k_1,V} \| \tilde{\pi}_r^{Med}), \quad \forall\, k \in \mathcal{C}_r^p. \tag{15}$$

With $\chi_r^{k,g}$ and $\chi_r^{k,Med}$, we design a joint maliciousness metric

$$\nu_r^k = -\chi_r^{k,g} + \epsilon \cdot \chi_r^{k,Med}. \tag{16}$$

Here, ϵ is a divergence weighting factor, which is a preset constant. Applying softmax on the joint metrics, the attention score for client $k \in \mathcal{C}_r^p$ is

$$\zeta_r^k = softmax(-\nu_r^k) = \frac{\exp^{-\nu_r^k}}{\sum_{k \in \mathcal{C}_r^p} \exp^{-\nu_r^k}}. \tag{17}$$

We aggregate the remaining variational parameters weighted by attention scores as the new global variational parameters:

$$U_r = \sum_{k \in \mathcal{C}_r^p} \zeta_r^k (\Gamma_r^{k,V} \circ U_r^{k,V}). \tag{18}$$

Essentially, the global variational approximation is denoted as

$$\theta_r \sim \tilde{\pi}_r^g = \mathcal{N}(U_r, \tilde{s}^2 I), \tag{19}$$

where θ_r denotes global weights in round r. In this way, FedBDA achieves resilient aggregation to guarantee the generalization and robustness of the global model.

Algorithm 2: Variational attention-based aggregation

Input: Rebuild local variational distributions $\{\tilde{\pi}_r^{k,V}|k\in\mathcal{C}_r\}$
Output: aggregated global variational parameters U_r

```
for k_1 ∈ C_r do
    for k_2 ∈ C_r \ k_1 do
        Calculate JS divergence χ_r^{k_1,k_2} between π̃_r^{k_1,V} and π̃_r^{k_2,V} via (11)
    end for
end for
C_r^1, C_r^2 ← arg min_{C_r^1 ∪ C_r^2 = C_r} ( max_{k_1∈C_r^1, k_2∈C_r^2} χ_r^{k_1,k_2} )
C_r^p ← arg max_{C∈{C_r^1, C_r^2}} |C|
Compute χ_r^{k,g} by (14) for any k ∈ C_r^p
Count the median of the remaining parameters and calculate χ_r^{k,Med} for any k ∈ C_r^p
Obtain joint metric ν_r^k ← −χ_r^{k,g} + ε · χ_r^{k,Med} for k ∈ C_r^p
Calculate attention scores based on joint maliciousness metrics for all clients in C_r^p via (17)
U_r ← weighted sum with attention scores via (18)
return U_r
```

VI. Analysis of FedBDA

A. Robustness Analysis

The robustness against Byzantine attacks has been defined in previous works [2], [17], as mentioned in Definition VI.1.

Definition VI.1 ((a,b)-robustness)**.** Supposed the proportion of malicious clients $a=\frac{K}{|\mathcal{A}|}\leq a_{max}\leq 0.5$, we are given local parameters $\{U^{k_1},\dots,U^{k_c}\}$ with a good subset $\mathcal{G}\subseteq\{k_1,\dots,k_c\}$ and $|\mathcal{G}|>(1-a)c$. For any $k_i,k_j\in\mathcal{G}$, there exists a constant $\ell>1$ such that

$$\mathbb{E}\|U^{k_i}-U^{k_j}\|^2\leq\ell^2. \tag{20}$$

Defining $\bar{U}=\frac{1}{|\mathcal{G}|}\sum_{k_i\in\mathcal{G}}U^{k_i}$, the output $\hat{U}$ of the robust aggregation satisfies

$$\mathbb{E}\|\hat{U}-\bar{U}\|^2\leq ab\ell^2. \tag{21}$$

Based on Definition VI.1, we derive the robustness of FedBDA. The good subset is denoted as $\mathcal{G}_r=\mathcal{C}_r\cap(\mathcal{K}\backslash\mathcal{A})$ in round r. The server measures JS divergences of pairwise local variation distributions. Let local variational parameters from benign clients satisfy

$$\mathbb{E}\left\|\Gamma_r^{k_i,V}\circ U_r^{k_i,V}-\Gamma_r^{k_j,V}\circ U_r^{k_j,V}\right\|^2\leq\ell^2,\ \forall k_i,k_j\in\mathcal{G}_r. \tag{22}$$

With $p=0.5$ and $\tilde{s}^2\leq 1$, following the basic techniques of [35], we can derive that

$$\begin{aligned}
&JS(\tilde{\pi}_r^{k_i,V}\|\tilde{\pi}_r^{k_j,V})\\
&\quad\leq S\log(N)+\frac{\left\|\Gamma_r^{k_i,V}\circ U_r^{k_i,V}-\Gamma_r^{k_j,V}\circ U_r^{k_j,V}\right\|^2}{8}\\
&\quad\leq S\log(2LD^2)+\frac{\ell^2}{8},\quad\forall k_i,k_j\in\mathcal{G}_r,
\end{aligned} \tag{23}$$

where the total parameter size satisfies $N\leq 2LD^2$ for the local model with L layers and D hidden sizes in each layer. If local variational distribution of client k_l exists $JS(\tilde{\pi}_r^{k_l,V}\|\tilde{\pi}_r^{k_i,V})>S\log(2LD^2)+\ell^2/8$, it must hold $k_l\in\mathcal{A}$. This is a sufficient condition, and the opposite direction may not necessarily hold. Our clustering strategy preserves benign updates and filters some poisonous updates with larger JS divergence differences (*i.e.*, larger than $S\log(2LD^2)+\ell^2/8$).

For the remaining client set $\mathcal{C}_r^p$ after clustering, there exists a constant $z\geq 1$ such that $\mathbb{E}\|\Gamma_r^{k_i,V}\circ U_r^{k_i,V}-\Gamma_r^{k_l,V}\circ U_r^{k_l,V}\|^2\leq z\ell^2$ for any $k_i,k_l\in\mathcal{C}_r^p$, where it is possible that $k_l\in\mathcal{B}_r$, and $\mathcal{B}_r$ denotes the malicious client set included in selected clients in round r with $\mathcal{B}_r=\mathcal{C}_r^p\cap\mathcal{A}=\mathcal{C}_r^p-\mathcal{G}_r$. The optimal attention factor for client $k_i\in\mathcal{G}_r$ is defined as $\zeta_r^{k,*}=\frac{1-a}{|\mathcal{G}_r|}$, where $|\mathcal{G}_r|>(1-a)c$ such that $|\mathcal{B}_r|=|\mathcal{C}_r^p|-|\mathcal{G}_r|\leq ac$. Referring to [2], we recall

$$\bar{U}_r=\frac{1}{|\mathcal{G}_r|}\sum_{k_i\in\mathcal{G}_r}\Gamma_r^{k_i,V}\circ U_r^{k_i,V}. \tag{24}$$

Based on the aggregated parameters $\hat{U}_r$ by (18), it holds that

$$\begin{aligned}
&\mathbb{E}\left\|\hat{U}_r-\bar{U}_r\right\|^2\\
&=\mathbb{E}\left\|\sum_{k_l\in\mathcal{C}_r^p}\zeta_r^{k,*}\left(\Gamma_r^{k_l,V}\circ U_r^{k_l,V}\right)-\frac{1}{|\mathcal{G}_r|}\sum_{k_i\in\mathcal{G}_r}\Gamma_r^{k_i,V}\circ U_r^{k_i,V}\right\|^2\\
&\leq 2a^2\mathbb{E}\left\|\frac{1}{|\mathcal{B}_r|}\sum_{k_l\in\mathcal{B}_r}\Gamma_r^{k_l,V}\circ U_r^{k_l,V}-\bar{U}_r\right\|^2\\
&\leq\frac{2a^2}{|\mathcal{B}_r|}\sum_{k\in\mathcal{B}_r}\left\|\Gamma_r^{k_l,V}\circ U_r^{k_l,V}-\bar{U}_r\right\|^2\leq 2az\ell^2,
\end{aligned} \tag{25}$$

Thus, we prove that our variational attention-based aggregation can achieve $(a,2z)$-robustness.

B. Convergence Analysis

The global variational approximation $\tilde{\pi}^g$ is explored in FedBDA to estimate the tempered posteriors of all client-side data. Sampling global weights $\theta\sim\tilde{\pi}^g$, we obtain the global model f_θ. Each device $k\in\mathcal{K}$ owns its benign data $\mathcal{D}^k\backslash D_{\mathcal{A}}^k$ and unknown true function f_0^k such that $y^k=f_0^k(\boldsymbol{x}^k)$ for any $(\boldsymbol{x}^k,y^k)\in\mathcal{D}^k\backslash D_{\mathcal{A}}^k$, where $D_{\mathcal{A}}^k=\emptyset$ if client k is not manipulated by the attacker. To analyze the convergence, we start with the following assumptions.

Assumption 1. The activation functions ϕ, ϱ, and ρ are 1-Lipschitz continuous.

Assumption 2. The absolute values of all weights in the optimal model θ^* have an upper bound $T\geq 2$.

Assumption 1 and 2 are common in Bayesian analysis works [35], [33] and they hold in the real world. The activation functions we used (*i.e.*, relu function ρ, tanh function ϱ, sigmoid function ϕ) are 1-Lipschitz continuous. The optimal global model is defined as $\theta^*=\arg\min_{\theta\in\mathcal{F}_{S,L,D}}\frac{1}{K}\sum_{k=1}^K\|f_\theta-f_0^k\|_2^2$, which is fixed. Hence, the absolute values of all weights in θ^* are fixed, and they must have an upper bound.

Definition IV.1 shows the generalization error of a Bayesian variational approximation. In FedBDA, we extend to the average generalization error of the global model $\theta\sim\tilde{\pi}^g$ across all client-side data, which can be defined as the expected average of the squared L2-distance between the global model f_θ and local true functions $\{f_0^k|k\in\mathcal{K}\}$, *i.e.*,

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^K\int\|f_\theta-f_0^k\|_2^2\tilde{\pi}^g(\mathrm{d}\theta)\right]. \tag{26}$$

To bound (26), we first introduce a specific posterior variance

$$\tilde{s}^2 = \frac{S}{16md^2}\log(3D)^{-1}(2TD)^{-2L}\Big\{\frac{1}{(2TD)^2-1} + \frac{2}{(2TD-1)^2} + \Big(d+1+\frac{1}{TD-1}\Big)^2\Big\}^{-1}, \quad (27)$$

where m is the number of samples participating in training. As mentioned in Section IV, the number of nonzero weights is represented by S. The input dimension is d, and we suppose that each hidden layer has D hidden units. Besides, L is the number of layers in DNN. With such a specific posterior variance, we derive the convergence of FedBDA.

Theorem 1. Let Assumptions 1 and 2 hold. Considering different numbers of local data in different clients such that the minimum amount of total input data up to round r is denoted as $m_r = r \times V \times \min\{|\mathcal{D}^1|, \ldots, |\mathcal{D}^K|\}$, for any $\alpha \in (0,1)$, the average generalization error of the global model f_θ learned by FedBDA satisfies:

$$\mathbb{E}\Big[\frac{1}{K}\sum_{k\in\mathcal{K}}\int \|f_\theta - f_0^k\|_2^2 \tilde{\pi}^g(\mathrm{d}\theta)\Big] \le \frac{2\sigma^2}{\alpha(1-\alpha)}(1+\frac{\alpha}{\sigma^2})\varepsilon_{m_r}^{S,L,D} + \frac{2}{K(1-\alpha)}\sum_{k=1}^{K}\xi^k \quad (28)$$

with

$$\varepsilon_{m_r}^{S,L,D} = \frac{ST^2}{2m_r} + \frac{SL}{m_r}\log(2TD) + \frac{3S}{m_r}\log(LD) + \frac{2S}{m_r}\log\Big(4d\max(\frac{m_r}{S},1)\Big), \quad (29)$$

and

$$\xi^k = \inf_{\theta^*\in\Theta_{S,L,D}} \|f_{\theta^*} - f_0^k\|_\infty^2, \quad (30)$$

where θ is the global weights sampled by (19) and σ^2 is a likelihood variance defined in Section IV-A.

The proof of Theorem 1 is shown in the appendix. Theorem 1 provides an upper bound of the average generalization error of the global model $\theta \sim \tilde{\pi}^g$, which can guarantee the convergence of FedBDA. Specifically, it involves two cases. **First**, if local true functions are actually neural networks with the structure of (S, L, D), the term ξ^k calculated by (30) is zero. Only the first term in (28) needs to be considered, in which $\varepsilon_{m_r}^{S,L,D}$ explicitly decreases with the increasing of m_r, according to (29). As training round r grows, the iterative training data amount m_r increases. Consequently, the upper bound of the generalization error reduces as the round grows, and FedBDA gradually converges.

On the other hand, if local true functions $\{f_0^k | k \in \mathcal{K}\}$ cannot be precisely represented by a neural network with the structure of (S, L, D), the error term ξ^k must be measured. Motivated by [33], [49], we assume that $\{f_0^k | k \in \mathcal{K}\}$ are β-Hölder smooth functions with $0 < \beta < d$. Based on basic techniques in [33] and Corollary 3 in [35], there exist constants C, C' such that $\xi^k \le C m_r^{\frac{-2\beta}{2\beta+d}}$ and $\varepsilon_{m_r}^{S,L,D} \le C' m_r^{\frac{-2\beta}{2\beta+d}} \cdot \log^2 m_r$. Hence, Inequality (28) holds that

$$\mathbb{E}\Big[\frac{1}{K}\sum_{k=1}^{K}\int \|f_\theta - f_0^k\|_2^2 \tilde{\pi}^g(\mathrm{d}\theta)\Big] \le C_1 m_r^{\frac{-2\beta}{2\beta+d}} \cdot \log^2 m_r, \quad (31)$$

where $C_1 > 0$ is a constant. Moreover, according to the minimax lower bound in Theorem 8 of [50], there exists a constant $C_2 > 0$ such that

$$\inf_\theta \frac{1}{K}\sum_{k=1}^{K}\int \|f_\theta - f_0^k\|_2^2 \tilde{\pi}^g(\mathrm{d}\theta) \ge C_2 m_r^{\frac{-2\beta}{2\beta+d}}. \quad (32)$$

The Inequality (31) bounds the generalization error for FedBDA with β-Hölder smooth functions, while Inequality (32) gives a minimax lower bound [50], [33] of the generalization error. Both (31) and (32) have the same term of $m_r^{\frac{-2\beta}{2\beta+d}}$, which indicates that the average generalization error of FedBDA converges at the *minimax optimal rate* $m_r^{\frac{-2\beta}{2\beta+d}}$ *up to a squared logarithmic factor* for the expected L_2-distance.

C. Overhead Analysis

Referring to [51], [52], we adopt Floating Point Operations (FLOPs) to characterize computation costs, which are correlated with the number of parameters updated during local training. Previous works [53], [54] have demonstrated that the dropout can reduce client-side FLOPs, where each client only updates nonzero parameters of the resulting model after dropout, contributing to mitigating local computation burden. For global aggregation, variational attention has a $\mathcal{O}(c^2N)$ time complexity, same as several advanced aggregations (*e.g.*, ClippedClustering [10] and FoolsGold [14]). Considering the powerful computing capability on the server, our variational attention will not delay the training process.

As for uplink communication costs, the uploaded parameter size is only determined by the dropout rate p. The larger dropout rate conduces to more dropped weight units and fewer uploaded parameters, bringing fewer uplink communication costs. However, a larger dropout rate is more likely to deteriorate model accuracy [55]. Therefore, it is crucial to choose an appropriate dropout rate for the trade-off between uplink cost reduction and model accuracy guarantee. According to historical experience in [56], [57], we generally adopt the dropout rate of 0.5 in this work, which provides $2\times$ reduction of uplink communication costs. In terms of downlink communication, the server transfers relatively dense global parameters to clients such that FedBDA has similar downlink costs as conventional FL frameworks (*e.g.*, FedAvg [1]).

VII. Experiments

A. Experiment Setup

Datasets. We consider four classic datasets for experiment evaluation, including MNIST [58], Fashion-MNIST (FMNIST) [59], CIFAR-10 [60], and Reddit [61]. Reddit is a federated benchmark for next-word prediction, which involves public comments posted on the social network. We adopt the top 100 users with more data as clients, where the distributed data are inherently non-IID. Other datasets are widely used for image classification. MNIST and FMNIST datasets contain 60,000 gray-level images in 10 object classes. For CIFAR-10, there are 60,000 color figures from 10 classes. We utilize the pathological partition [53] to simulate non-IID data, where each client is randomly assigned 2 classes of data.

TABLE I
Accuracy and uplink cost results for various Byzantine attacks on MNIST and FMNIST.

Methods	MNIST						FMNIST					
	ALIE	IPM	SF	Mimic	LF	UpComm	ALIE	IPM	SF	Mimic	LF	UpComm
Median	67.35±1.56	79.69±1.51	64.61±2.59	83.30±1.60	74.44±1.09	0.83MB	62.83±2.55	69.22±1.45	67.69±1.99	67.20±1.93	71.28±1.31	7.76MB
Trim-Mean	78.39±2.27	90.38±0.72	88.41±1.56	88.03±1.61	83.80±1.23	0.83MB	69.12±1.39	76.46±1.25	74.54±1.37	73.77±0.89	72.80±1.25	7.76MB
RFA	86.51±1.00	93.16±0.57	88.67±1.48	90.98±0.95	89.77±0.78	0.83MB	72.47±0.68	76.40±0.89	76.28±1.31	74.53±0.95	75.17±1.43	7.76MB
Krum	42.78±3.10	24.54±4.43	34.90±2.47	42.16±2.42	16.57±3.29	0.83MB	32.88±1.54	34.02±1.37	41.05±2.57	35.67±1.49	37.61±3.29	7.76MB
Bulyan	86.35±2.07	91.44±1.20	91.01±1.11	87.90±0.59	91.17±0.71	0.83MB	70.78±1.58	70.56±0.73	72.50±1.25	71.32±0.58	73.84±1.38	7.76MB
FoolsGold	89.07±0.85	88.62±0.78	86.58±1.37	88.03±0.98	85.39±0.88	0.83MB	74.82±1.00	75.32±1.10	72.76±0.53	75.44±0.98	71.62±0.47	7.76MB
CC	92.33±1.82	92.41±0.44	91.82±0.86	93.36±0.69	93.80±0.44	0.83MB	73.04±1.22	72.10±0.71	71.80±0.93	72.24±0.82	72.79±0.81	7.76MB
ClippedCluster	81.29±1.48	88.20±1.07	86.33±1.39	87.55±1.76	88.55±0.84	0.83MB	71.98±1.94	75.11±1.57	73.34±0.86	71.95±1.13	75.89±1.27	7.76MB
Bucketing	90.20±1.18	93.51±1.37	92.74±1.47	93.41±1.58	92.19±1.05	0.83MB	69.70±0.60	70.58±1.28	72.40±0.62	73.18±0	71.80±0.77	7.76MB
GAS	89.03±0.82	92.96±0.85	90.47±0.84	90.89±0.61	93.92±0.40	0.83MB	74.46±0.68	76.09±0.76	73.71±0.94	74.72±1.78	75.20±1.25	7.76MB
BayBFed	83.62±0.98	93.89±0.97	93.43±0.63	94.15±0.62	93.95±1.05	0.83MB	72.93±0.96	73.71±0.91	75.28±1.61	74.54±1.11	73.42±1.18	7.76MB
FLAP	86.33±0.75	88.71±1.09	93.25±0.72	91.91±0.97	91.31±0.92	0.80MB	73.83±1.13	75.13±0.83	76.49±1.03	73.54±0.79	73.85±0.75	7.76MB
FedREP	90.64±1.07	92.31±0.77	91.32±0.69	92.13±0.60	93.43±0.46	0.43MB	72.42±0.95	71.14±0.79	72.11±1.31	73.75±1.37	73.81±0.63	3.94MB
FedBDA	**95.36**±0.40	**95.22**±0.33	**95.18**±0.59	**95.40**±0.37	**95.02**±0.14	**0.43MB**	**76.08**±0.95	**78.74**±0.65	**77.67**±0.82	**77.42**±1.07	**78.86**±0.71	**3.94MB**

TABLE II
Accuracy and uplink cost results for various Byzantine attacks on CIFAR-10 and Reddit.

Methods	CIFAR-10						Reddit					
	ALIE	IPM	SF	Mimic	LF	UpComm	ALIE	IPM	SF	Mimic	LF	UpComm
Median	30.54±1.28	26.38±1.78	33.24±2.14	35.75±0.16	33.97±1.49	30.44MB	30.87±0.11	30.29±0.24	29.44±0.17	29.69±0.17	30.58±0.30	284.38MB
Trim-Mean	32.18±1.58	43.27±1.35	42.22±1.14	40.33±1.06	39.53±1.93	30.44MB	31.39±0.29	30.61±0.42	29.56±0.42	30.32±0.23	31.11±0.36	284.38MB
RFA	42.93±1.37	41.03±1.40	43.57±1.70	40.51±1.40	42.99±1.31	30.44MB	31.27±0.45	30.60±0.40	29.35±0.46	30.17±0.24	31.01±0.28	284.38MB
Krum	12.35±1.02	11.10±0.32	18.86±1.50	16.58±0.98	14.44±1.04	30.44MB	23.00±0.56	23.52±0.49	22.96±0.60	24.47±0.23	22.98±0.59	284.38MB
Bulyan	15.78±3.02	15.40±2.66	19.30±1.11	17.06±2.01	17.76±2.15	30.44MB	30.90±0.09	30.22±0.21	29.62±0.33	29.61±0.31	29.61±0.37	284.38MB
FoolsGold	42.77±0.67	43.44±2.93	43.39±1.06	38.59±1.07	43.60±1.76	30.44MB	30.27±0.41	29.22±0.18	28.24±0.33	29.06±0.13	29.46±0.30	284.38MB
CC	35.71±1.07	30.78±1.42	33.14±0.64	35.81±1.06	33.39±1.56	30.44MB	31.38±0.70	31.07±0.06	30.90±0.15	31.13±0.11	31.11±0.04	284.38MB
ClippedCluster	40.68±0.96	41.07±1.21	43.42±1.85	41.01±1.39	41.35±2.56	30.44MB	31.44±0.56	30.24±0.25	30.34±0.51	30.21±0.18	30.89±0.28	284.38MB
Bucketing	34.66±2.62	32.42±1.90	32.10±1.19	33.78±1.28	31.57±1.23	30.44MB	31.39±0.07	31.08±0.05	31.26±0.12	31.20±0.04	31.65±0.06	284.38MB
GAS	43.53±1.92	43.35±1.80	44.14±1.36	42.84±0.78	43.37±1.88	30.44MB	30.61±0.34	30.03±0.21	30.64±0.30	29.92±0.32	30.61±0.33	284.38MB
BayBFed	40.97±1.36	38.97±1.24	42.57±1.25	42.33±2.58	42.76±2.09	30.44MB	31.10±0.14	30.19±0.21	29.18±0.41	30.13±0.30	30.93±0.17	284.83MB
FLAP	38.87±0.95	41.63±0.84	43.86±0.81	41.48±1.24	40.64±1.41	30.44MB	31.67±0.11	30.18±0.12	29.28±0.57	29.98±0.19	30.74±0.48	284.38MB
FedREP	37.77±0.82	40.41±0.97	40.62±0.82	41.16±1.38	40.14±1.26	15.36MB	30.94±0.20	30.42±0.04	29.75±0.36	30.34±0.29	30.99±0.15	227.16MB
FedBDA	**46.56**±0.88	**45.94**±0.82	**45.48**±1.15	**44.10**±0.51	**46.08**±0.85	**15.36MB**	**33.16**±0.06	**33.36**±0.23	**32.98**±0.15	**33.09**±0.20	**33.34**±0.21	**156.17MB**

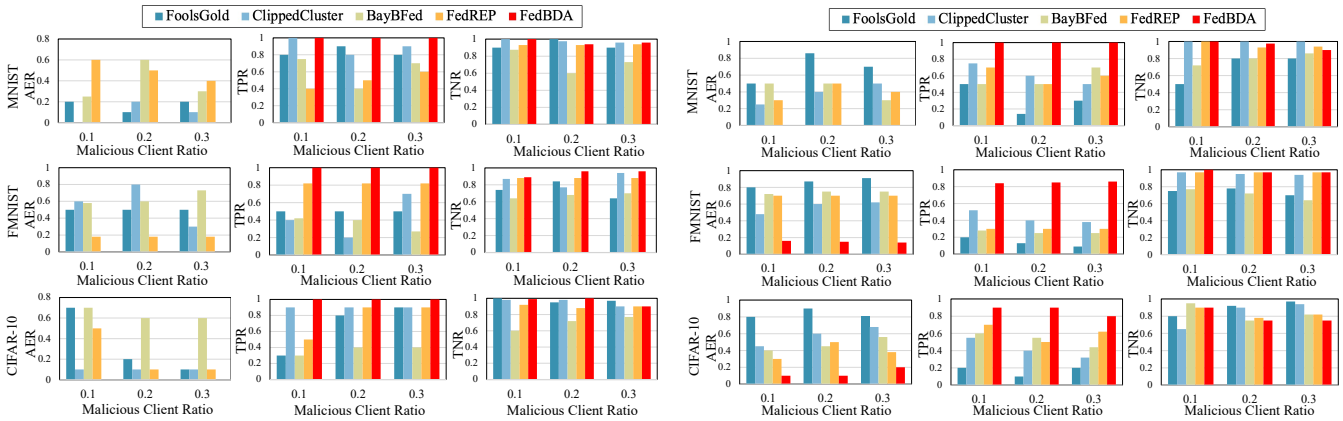
Models. CNN-based classification models are used for MNIST and CIFAR-10, referring to [62], [63]. A two-layer Fully-Connected Neural Network (FCNN) is trained on FMNIST. Following [61], we adopt a RNN model with two LSTM layers for Reddit, where a vocabulary with 10,000 words is built.

Implementation Settings. In our experiments, there are 100 clients for all datasets, and the proportion of malicious clients is 10% by default. In each round, the server randomly selects $c = 10$ clients to participate in training. We set a uniform dropout rate $p = 0.5$ for all clients. During local training, the batch size is set to 20, and the loss gap is evaluated every $\tau = 3$ iterations for Reddit. In terms of image datasets, each client assesses the loss gap every five iterations with a batch size of 32. We set the total round to $R = 100$ on three image datasets, while $R = 60$ is adopted for Reddit. The boundary of two stages in Bayesian adaptive dropout is set to $\tau_R = 55$ for Reddit and $\tau_R = 90$ for other datasets.

Byzantine Attacks. We consider five Byzantine attacks.

- *ALIE*. The attacker counts the mean $\bar{U}$ and standard deviation $\bar{\sigma}$ of benign parameters, and poisons clients by sampling model parameters from $(\bar{U} - z_1\bar{\sigma}, \bar{U} + z_1\bar{\sigma})$.
- *IPM*. The model parameters of malicious clients are tampered by $-\frac{z_2}{|\mathcal{G}_r|}\sum_{k_i \in \mathcal{G}_r} U^{k_i}$ with $z_2 = 0.1$.
- *SF*. The signs of all local model parameters in malicious clients are simply flipped.
- *Mimic*. The attacker picks a good client and copies its model parameters to malicious clients.
- *LF*. The data labels of malicious clients are flipped by $F(y) := C_D - y$, where C_D is the number of classes.

For implementation details, Byzantine attacks are injected into malicious clients in each round. Due to the random client selection, a portion of malicious clients are selected to participate in training in each round. Referring to [16], [17], [64], attacked operations and target clients remain unchanged throughout the FL process. For the malicious client, the first four model parameter-based attacks (*i.e.,* ALIE, IPM, SF, and Mimic) poison parameters once in a random iteration during local training. After local training, the same attack is injected again before parameter transmission, following [10], [17]. The final local parameters are partially masked based on the dropping pattern and then transmitted in a sparse form. Data-based LF attack relabels a data class of malicious clients, and these wrong data are used to train in the FL process [65].



(a) The evaluation results of AER, TPR, TNR for FedBDA with ALIE.

(b) The evaluation results of AER, TPR, TNR for FedBDA with LF.

Fig. 3: The attack resistance results about AER, TPR, TNR.

Baselines. We compare against a variety of robust FL frameworks, including (1) statistical aggregations, (2) distance-based defenses, and (3) combined strategies with auxiliary defenses and resilient aggregations. For *statistical aggregations*, we consider *Median*, *Trim-Mean* [12], and *RFA* [66], which calculate the coordinate-wise median, trimmed mean, and geometric median of all local updates to estimate new global parameters. For *distance-based defenses*,

- *Krum* [13] and *Bulyan* [8] are typical distance-based aggregations, which detect and penalize some malicious clients according to Euclidean distances.
- *FoolsGold* [14] identifies malicious updates based on cosine similarity of pair-wise local updates.
- *Centered Clipping (CC)* [2] clips local updates based on L_1 distances between local and global model parameters.
- *ClippedCluster* [10] integrates local update clipping and cosine similarity-based clustering.

In terms of *combined strategies*, we compare against

- *Bucketing* [17] locally weighs current updates with the updates in the last round, globally partitions all updates into several buckets, and aggregates the averages of every bucket by existing robust rules (*e.g.*, CC).
- *GrAdient Splitting (GAS)* [28] splits local gradients into sub-vectors and detects malicious gradients by evaluating L_1 distances between sub-vectors.
- *BayBFed* [18] filters out malicious updates by drawing probabilistic distributions of local updates and adapting the Chinese Restaurant Process to JS divergences between pairwise distributions.
- *FLAP* [32] zeros out partial weight units of the global model after server-side robust aggregation.
- *FedREP* [31] introduces magnitude-based unstructured sparsification into local models, and sparse updates are aggregated with existing resilient rules on the server.

Evaluation Metrics. In our experiments, test accuracy is used as a main metric, which can show the overall learning performance under Byzantine attacks [11], [67]. Referring to [18], [68], we further introduce three metrics to intuitively reflect the ability to resist attacks, including: **(1) Attack Escape Rate (AER)** [68] indicates the rate at which malicious clients escape from the defense method, equivalent to the success rate of attacks [64]. A smaller AER means stronger attack resistance. **(2) True Positive Rate (TPR)** [18] indicates how accurately the defense resists attacks. The total number of correctly recovered/detected malicious clients is called True Positives (TP), and the number of malicious clients judged to be benign is called False Negatives (FN). TPR = TP / (TP + FN). **(3) True Negative Rate (TNR)** [18] indicates the ability to identify benign clients. The total number of correctly identified benign clients is called True Negatives (TN), and the number of benign clients detected to be malicious is called False Positives (FP). TNR = TN / (TN + FP).

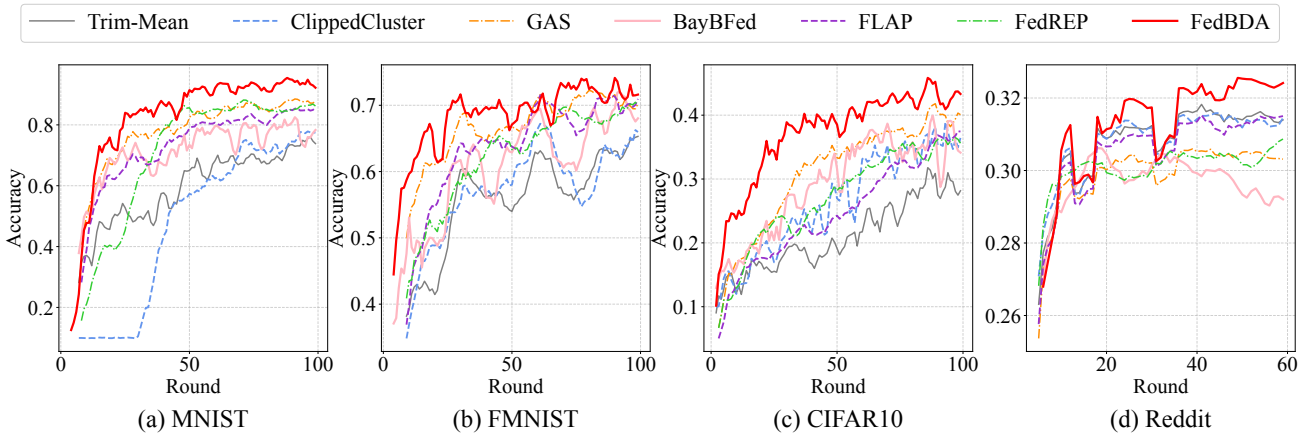
TABLE III
Malice resistance degree of two modules.

MCR	MNIST		FMNST		CIFAR-10	
	BAD	VAA	BAD	VAA	BAD	VAA
ALIE-0.1	80%	20%	75%	25%	86%	14%
ALIE-0.2	75%	25%	88%	12%	80%	20%
ALIE-0.3	95%	5%	85%	15%	95%	5%
ALIE-0.4	48%	42%	87%	10%	83%	15%
LF-0.1	80%	20%	65%	18%	80%	10%
LF-0.2	90%	10%	80%	5%	73%	18%
LF-0.3	90%	10%	80%	5%	72%	8%
LF-0.4	80%	2%	65%	12%	52%	23%

B. Experimental Results

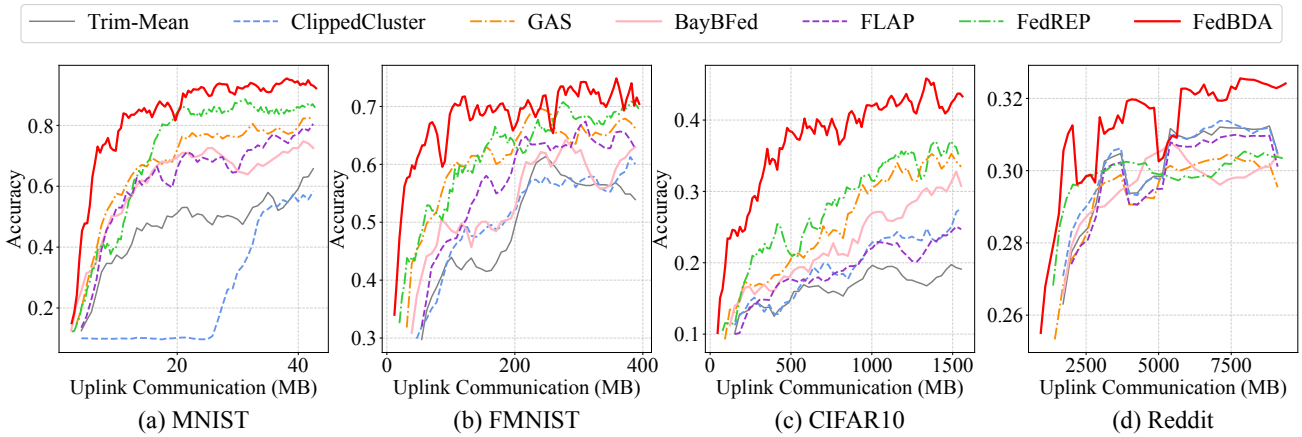
The test accuracy and uplink communication results of four datasets on various attacks are listed in Table I and II. For the next-word prediction task on Reddit, we adopt the top-3 accuracy to evaluate global benign performance, since mobile keyboards generally include three candidates, as mentioned in [69]. We observe that FedBDA consistently achieves state-of-the-art accuracy with fewer uplink communication costs on four datasets with different Byzantine attacks.

First, statistical aggregations Median and Trim-Mean generally offer poor performance on image classification tasks,



(a) MNIST (b) FMNIST (c) CIFAR10 (d) Reddit

Fig. 4: Test accuracy versus rounds on four datasets with ALIE attack.



(a) MNIST (b) FMNIST (c) CIFAR10 (d) Reddit

Fig. 5: Test accuracy versus total uplink communication costs on four datasets with ALIE attack.

especially under ALIE and mimic attacks. It implies that simple median and mean aggregation can be bypassed by some Byzantine attacks. RFA explores the geometric median through alternating minimization, which typically shows higher accuracy than Median and Trim-Mean. Compared to RFA, FedBDA presents significant benefits in accuracy and uplink efficiency, which improves 1.39%-8.85% accuracy with 2× uplink cost reduction on four datasets under ALIE attack.

Second, two typical distance-based methods Krum and Bulyan show poor performance in non-IID settings. With cosine similarity of pairwise local updates, FoolsGold outperforms Krum and Bulyan, which has the second-highest accuracy on CIFAR-10 with IPM and LF attacks. Compared to FoolsGold, FedBDA achieves up to 9.31% accuracy gains while reducing 2× uplink costs. Besides, parameter magnitude clipping is proposed in CC, and ClippedCluster further integrates it with cosine similarity-based clustering. We notice that ClippedCluster generally provides higher accuracy than CC on CIFAR-10 and Reddit, while FedBDA outperforms ClippedCluster in both test accuracy and uplink costs.

Furthermore, although Bucketing and GAS integrate auxiliary defenses with robust aggregations to alleviate the non-IID problem, they still are not enough to solve the performance degradation problem and offer lower accuracy in our highly non-IID settings, as shown in Tables I and II. FedBDA outperforms Bucketing and GAS in accuracy and uplink efficiency, which achieves 1.1%-6.33% accuracy improvements while reducing 2× uplink costs compared to GAS. With probabilistic measures of local updates, BayBFed provides the second-largest accuracy on MNIST with four attacks. Compared with BayBFed, FedBDA provides 1.66%-11.74% accuracy gains with 2 × uplink cost reduction. Moreover, FLAP and FedREP introduce dropout into robust FL. FLAP prunes the global model in the server, which only affects downlink overhead without involving more constrained uplinks. FedREP sparsifies client-side model parameters and is applied to fully connected and convolutional layers in [31] without considering recurrent layers. Thus, for image classification tasks with CNNs or

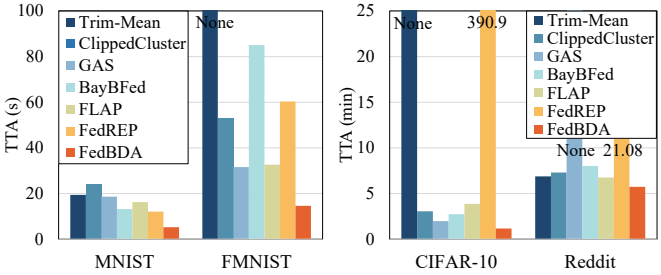
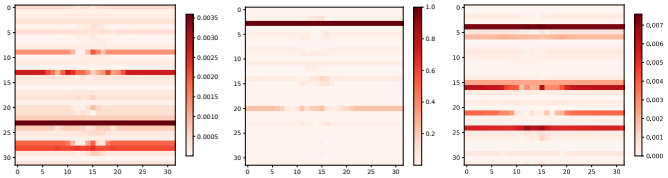


Fig. 6: Time to Target Accuracy (TTA) evaluations.

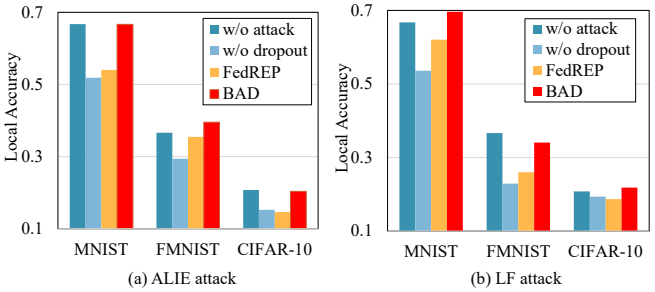
TABLE IV
Ablation experiment results.

Methods	MNIST	FMNIST	CIFAR-10	Reddit
FedAvg	87.84±0.88	74.34±1.22	42.91±1.18	29.31±0.62
B-FedAvg	93.30±0.70	75.66±2.05	44.24±1.73	29.61±0.17
B-VAA	**95.41±0.59**	76.29±1.84	45.15±1.48	29.99±0.39
BD-FedAvg	94.83±0.23	76.42±0.73	44.36±1.06	31.59±0.12
FedBDA	95.18±0.59	**77.67±0.82**	**45.48±1.15**	**32.98±0.15**



(a) without attack (b) with LF attack (c) BAD with LF attack

Fig. 7: The local update distribution for the malicious client.



(a) ALIE attack (b) LF attack

Fig. 8: Test accuracy of local models for the malicious client.

FCNN, FedBDA does not significantly outperform FedREP in uplink costs, but it increases accuracy by 1.59%-8.79% compared to FedREP. For Reddit with LSTM models, FedBDA can accommodate recurrent connections, which brings 1.5 × uplink reduction compared to FedREP. Meanwhile, FedBDA offers up to 2.35% accuracy gains on Reddit.

Resistance. To intuitively reflect the ability to resist attacks, we evaluate AER, TPR, and TNR under different malicious client ratios. Note that if the test accuracy of the local model trained with Bayesian Adaptive Dropout (BAD) under attacks is better than the received global model, the malicious client is considered to be successfully resisted during local training. As for calculating the three metrics, FedBDA integrates local attack resistance of BAD and global resistance of Variational Attention-based Aggregation (VAA). Similarly, FedREP jointly considers the attack resistance of local magnitude-based dropout and global resilient aggregation with distance-based clustering. As shown in Figure 3, FedBDA consistently outperforms baselines in terms of AER and TPR, which overcomes all malicious clients in most cases, achieving $AER = 0$ and $TPR = 1$. For TNR, FedBDA generally provides higher TNR, sometimes reaching $TNR = 1$. Compared to ClippedCluster, FedBDA can decrease 10%-80% AER and improve up to 80% TPR with up to 19% TNR gain. Besides, FedBDA achieves 10%-60% reduction in AER and up to 60% gain in TPR, compared to FedREP. With the increase of the malicious client ratio, FedBDA still effectively resists the Byzantine attack and maintains the lowest AER and highest TPR.

Furthermore, we analyze the resistance degree of the two modules of FedBDA (*i,e.*, BAD and VAA) separately. Table III shows the results of the two modules under ALIE and LF attacks. If the test accuracy of the uploaded local model is better than that of the original global model, we consider that the malice is successfully resisted by BAD. For ALIE attacks, BAD removes malicious units with the benignity-guided dropping pattern when the attacker injects the attack during local training. Even if the attack is injected again before transmission, the highly malicious units can also be promptly dropped based on the benignity-guided pattern, so that the model accuracy will not fluctuate greatly, guaranteeing the success of BAD. For data-based LF, BAD can mitigate model shifts caused by the poisoned data during client-side training to ensure local robustness. The reason why the sum of BAD and VAA terms is not up to 1 is that FedBDA cannot resist all malicious clients (*i.e.*, $AER > 0$) in a few cases. By observing the experimental results, we get that local BAD can catch most malicious attacks, the remaining small amount of malicious clients are resisted by global VAA. For instance, on MNIST with 10% malicious clients, BAD can cover 80% malicious attacks, while the remaining 20% is defended by VAA.

Convergence. To evaluate the convergence of FedBDA, we report the test accuracy varying with global rounds and total uplink costs. We observe from Fig. 4 that FedBDA quickly converges to higher accuracy than baselines. For example, FedBDA reaches up to 46% accuracy on CIFAR-10 with the ALIE attack after 100 rounds, while ClippedCluster and FedREP offer less than 40% accuracy. Besides, FedBDA consistently provides the highest accuracy under the same uplink costs, as shown in Fig. 5. For MNIST, FedBDA reaches 95.76% with 40MB uplink costs, while FLAP and FedREP only offer 80.39% and 89.43% accuracy, respectively.

For further quantifying efficiency advantages of FedBDA, we adopt Time to Target Accuracy (TTA) [70] to characterize the total running time reaching target accuracy. TTA results are shown in Fig. 6, where 'None' means that the method cannot achieve target accuracy. 81%, 72%, 38%, and 31% are preset target accuracy for MNIST, FMNIST, CIFAR-10, and Reddit. We notice that Trim-Mean cannot reach the preset target accuracy for FMNIST and CIFAR-10, and BayBFed is unable to obtain 32% accuracy on Reddit. Although FedREP alleviates the uplink burden, it utilizes unstructured sparsification and needs to traverse all model weights, leading to larger time delays for complicated models. Thus, FedREP shows the larger time costs on CIFAR-10 and Reddit with complex models. It can be seen that FedBDA always takes the shortest time to obtain target accuracy. For CIFAR-10, FedBDA reaches 38% after 70.15s, which reduces more than 40% (vs. 12.83s) time delay compared to the state-of-the-art method. For MNIST, FedBDA saves 56.9% time costs compared to baselines.

C. Ablation Studies

We verify the effectiveness of two core modules (*i.e.*, Bayesian adaptive dropout and variational attention-based

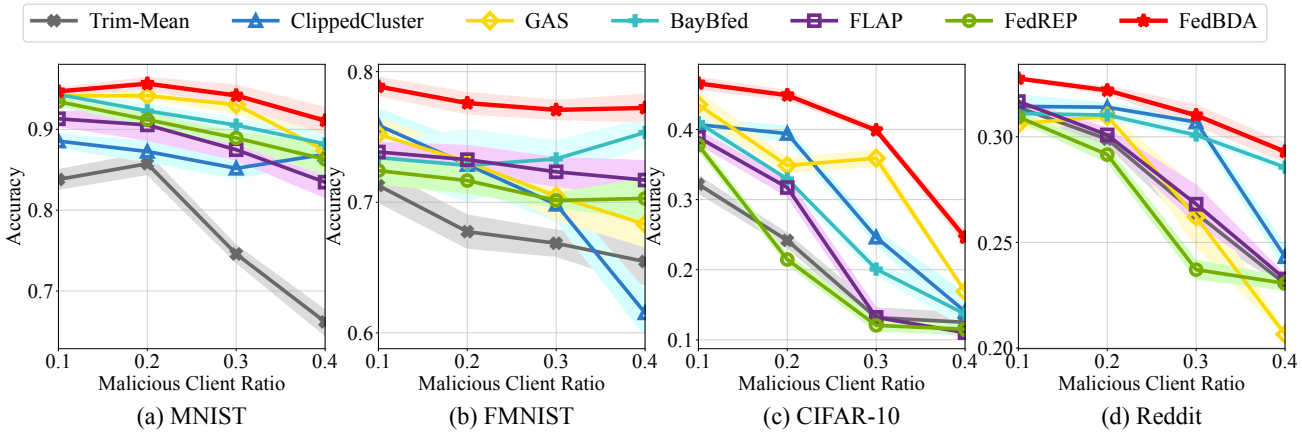


Fig. 9: Test accuracy versus malicious client ratios on four datasets.

aggregation) in FedBDA. First, we evaluate the impact of Bayesian models under FedAvg. Table IV shows the accuracy results of conventional FedAvg (with fixed model parameters) and FedAvg with Bayesian models (*i.e.*, B-FedAvg) where model parameters are viewed as random variables. We observe that B-FedAvg achieves significant accuracy gains on MNIST and FMNIST compared to FedAvg. In particular, B-FedAvg improves accuracy by more than 5% on MNIST.

Effect of Benignity Indicator-based Bayesian Dropout. Without dropout, each client directly trains a dense Bayesian model and the server aggregates local updates with variational attention (*i.e.*, B-VAA). As shown in Table IV, for MNIST, FedBDA reduces 2× uplink costs with the accuracy guarantee compared with B-VAA. For the other three datasets, FedBDA outperforms B-VAA in accuracy and uplink costs, indicating the effectiveness and efficiency of Bayesian adaptive dropout.

To further verify the local success of BAD , we compare the partial parameter distributions of local updates in the malicious client. As shown in Figure 7, the LF attack causes local updates to severely deviate from the original distribution without the attack. While BAD with benignity indicator effectively addresses this deviation and brings local updates closer to the original distribution, which proves that BAD can remove malicious weight units to maintain a benign model distribution, thereby mitigating local performance loss. Quantitatively, we evaluate the local model accuracy of the malicious client on the test dataset to demonstrate the performance advantage of the BAD. As shown in Figure 8, without dropout, LF and ALIE attacks severely degrade local accuracy. The magnitude-based dropout in FedREP slightly alleviates the loss, but the local accuracy is still much lower than the cases without the attack (*i.e.*, w/o attack). Under the Byzantine attack, BAD can generally recover the local accuracy to the same level as the unattacked case, or even slightly improve it, which indicates the effectiveness of our Bayesian dropout.

Effect of Variational Attention-based Aggregation. With Bayesian models, we compare the test accuracy of B-FedAvg and B-VAA, where only the global aggregation strategies are different. It can be seen from Table IV that B-VAA consistently outperforms B-FedAvg in terms of test accuracy on four datasets, demonstrating the advantages of our variational attention-based aggregation. Furthermore, with Bayesian adaptive dropout, we also evaluate the performance of average aggregation (*i.e.*, BD-FedAvg) and variational attention-based aggregation (*i.e.*, FedBDA). We observe that BD-FedAvg always shows lower test accuracy than FedBDA across four datasets, implying that our variational attention-based aggregation effectively enhances the robustness of the global model.

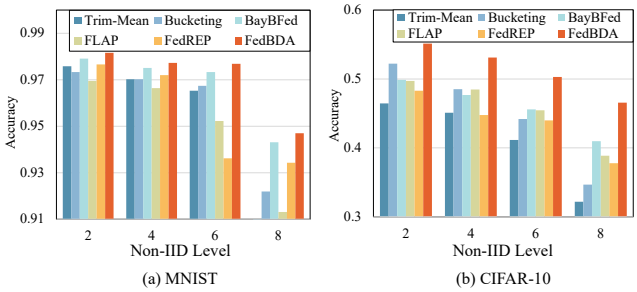


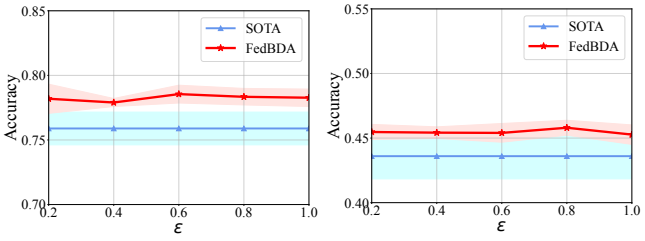
Fig. 10: Test accuracy on different non-IID levels.

D. Hyperparameters Test

We analyze the impact of several hyperparameter settings, including malicious ratios, non-IID levels, divergence weighting factor, and dropout rates.

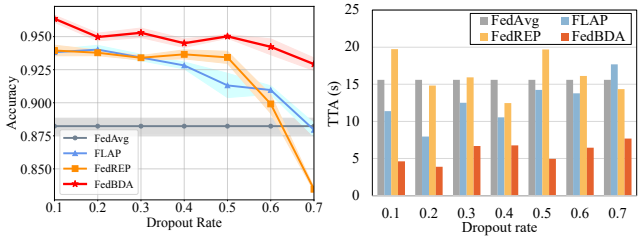
Malicious Client Ratios. Fig. 9 shows the accuracy results under different malicious ratios. We observe that FedBDA keeps the best performance under different malicious ratios, compared to other baselines. Generally, as more malicious clients are injected, the global model accuracy declines. Trim-Mean performs worst on MNIST and FMNIST with below 83% and 71% accuracy. Although FedREP mitigates uplink overhead, it provides the poorest performance for CIFAR-10 and Reddit. FedBDA achieves apparent accuracy gains on FMNIST and CIFAR-10, bringing up to 10% improvement compared to the state-of-the-art baseline.

Non-IID Levels. The accuracy results of different non-IID levels are reported in Fig. 10, where non-IID levels are characterized by the missing number of classes in the training data, and the more lacking classes indicate the higher non-IID degrees. As shown in Fig. 10, FedBDA consistently

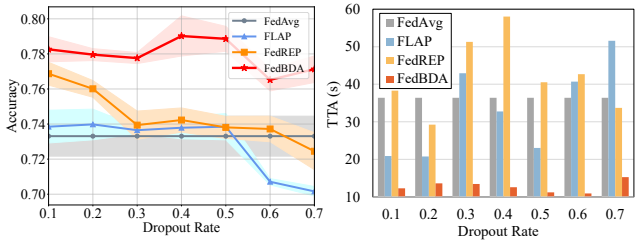


(a) FMNIST (b) CIFAR-10

Fig. 11: Test accuracy versus divergence weighting factors ϵ.



(a) Accuracy and TTA versus dropout rates on MNIST.



(b) Accuracy and TTA versus dropout rates on FMNIST.

Fig. 12: Accuracy and TTA versus dropout rates.

outperforms baselines under different non-IID levels. The accuracy gains are more obvious on CIFAR-10, where FedBDA enhances accuracy by 2.4%-5.6% compared to the state-of-the-art baseline. The colorful CIFAR-10 images contain more complicated features than gray-level images of MNIST, thus FedBDA performs better on complex datasets. The global performance generally decreases with the increase of non-IID levels. While the accuracy advantage of FedBDA becomes apparent as the non-IID level rises, especially on CIFAR-10.

Divergence Weighting Factor ϵ. In (16), divergence weighting factor ϵ is introduced to obtain the joint maliciousness metric, which affects the attention scores of local updates. We evaluate test accuracy under different divergence weighting factors $\epsilon \in \{0.2, 0.4, 0.6, 0.8, 1\}$. As shown in Figure 11, different choices of ϵ have a little fluctuation in accuracy. Empirically, we recommend setting $\epsilon \in [0.6, 0.8]$. Notably, the test accuracy of FedBDA is consistently better than that of the state-of-the-art method under different divergence weighting factors, proving the robustness of FedBDA.

Dropout Rates. So far, there have been many empirical discussions about dropout rate settings [55], [56], [57], [71]. Following [57], [71], the dropout rate should be set empirically between 0.1 and 0.6 for the trade-off between accuracy and efficiency. We evaluate the test accuracy and Time to Target Accuracy (TTA) for different dropout rates $p \in \{0.1, 0.2, \ldots, 0.7\}$ on MNIST and FMNIST with the LF attack. As shown in Fig. 12, accuracy results of FedREP and FLAP generally trend downward as the dropout rate increases. With $p \leq 0.5$, FedREP and FLAP outperform FedAvg. However, the performance of FLAP significantly drops at $p > 0.5$, even 3.15% lower than FedAvg on FMNIST. FedREP shows 0.86% accuracy loss at $p = 0.7$. As for time delay, FedREP and FLAP take more time to reach the target accuracy than FedAvg at some dropout rates (*e.g.*, $p = 0.3$ and $p = 0.6$ on FMNIST). Noticeably, FedBDA can provide the highest accuracy with the least time costs compared to baselines, demonstrating the effectiveness and stability of FedBDA.

Furthermore, FedMP [55] proposes a Multi-Armed bandit-based online learning strategy to adaptively determine dropout rates, which is orthogonal to our work and can be directly applied to FedBDA. As shown in Table V, we compare the fixed setting with $p = 0.5$ and the adaptive strategy in FedMP under the FedBDA framework. The adaptive strategy slightly improves accuracy on MNIST and CIFAR-10 with more uplink costs. For FMNIST and Reddit, the fixed setting with $p = 0.5$ performs better than the adaptive strategy.

TABLE V
The comparison of dropout rate strategies.

Dataset	Fixed ($p = 0.5$)		Ada ($p \in [0.1, 0.6]$)	
	Acc (%)	UpComm	Acc (%)	UpComm
MNIST	95.02±0.14	43MB	95.12±0.28	59MB
FMNIST	78.86±0.71	394MB	77.45±0.50	565MB
CIFAR-10	46.08±0.85	1536MB	46.14±0.74	2193MB
Reddit	33.34±0.21	9.15GB	33.22±0.28	9.38GB

E. Discussion

The variational attention-based aggregation in FedBDA has a computational complexity of $O(c^2N)$, where c is the number of selected clients in a round and N is the parameter size. This quadratic complexity can be alleviated by approximate attention to make our framework more feasible for large-scale FL. We explore a simpler approximate attention mechanism, Global-Median Divergence-based weighted Aggregation (GMDA), which removes the clustering step of FedBDA. For c local updates of the selected clients in a round, GMDA integrates their JS divergences with the global distribution and with the median distribution into the joint metric in (16) as the aggregation weight. In this way, the computational complexity of the GMDA is reduced to $\mathcal{O}(cN)$, where N is the number of global variational parameters. As shown in Table VI, GMDA performs slightly worse than Variational Attention-based Aggregation (VAA) in most cases, but it always outperforms other baselines as listed in Table I and Table II. Thus, GMDA with a $\mathcal{O}(cN)$ complexity can be considered in large-scale FL deployments. Our proposed VAA generally achieves the highest accuracy, which is a better choice in scenarios with strict accuracy requirements.

VIII. Conclusion

In this paper, we propose a novel FL robust framework, termed FedBDA, which takes the first step to locally quantify

TABLE VI
Accuracy results of the approximate attention and our variational attention aggregations.

Attacks	MNIST		FMNST		CIFAR-10		Reddit	
	GMDA	VAA	GMDA	VAA	GMAD	VAA	GMDA	VAA
ALIE	93.12±0.40	**95.36±0.40**	**76.99±0.68**	76.08±0.95	45.42±0.82	**46.56±0.88**	**33.41±0.15**	33.16±0.06
IPM	94.01±0.46	**95.22±0.33**	78.39±0.69	**78.74±0.65**	45.43±0.74	**45.94±0.82**	33.01±0.22	**33.36±0.23**
SF	92.68±0.69	**95.10±0.59**	77.05±0.67	**77.67±0.82**	44.61±0.79	**45.48±1.15**	32.86±0.24	**32.98±0.15**
Mimic	94.25±0.50	**95.40±0.37**	**77.50±0.41**	77.42±1.07	44.09±0.76	**44.10±0.51**	32.71±0.16	**33.09±0.20**
LF	94.70±0.91	**95.02±0.14**	77.63±0.63	**78.86±0.71**	45.04±1.12	**46.08±0.85**	33.28±0.30	**33.34±0.21**

weight units-wise benign scores and enables dual local-global robustness guarantee with theoretical Bayesian interpretation in non-IID settings. Specifically, we introduce variational Bayesian inference into local models to characterize dropout using spike-and-slab distributions. Each client independently maintains a unit-wise benignity indicator according to local performance changes, and adaptively drops poisonous and insignificant units of local variational distributions for client-side robust training. Furthermore, a variational attention scheme is designed to globally detect the potential maliciousness of local variational distributions based on joint metrics of JS divergence among local, global, and median distributions for resilient weighted aggregation. The effectiveness and efficiency of FedBDA are demonstrated through formal theoretical analysis and extensive experiments on four classic datasets.

ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China (2021YFB2900102), National Natural Science Foundation of China (62472410 and 62072436), and National Science Fund for Distinguished Young Scholars of China (62425201).

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[2] S. P. Karimireddy, L. He, and M. Jaggi, "Learning from history for byzantine robust optimization," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[3] B. Zhao, P. Sun, T. Wang, and K. Jiang, "FedInv: Byzantine-robust federated learning by inversing local model updates," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[4] Z. Gong, L. Shen, Y. Zhang, L. Y. Zhang, J. Wang, G. Bai, and Y. Xiang, "AgrAmplifier: Defending federated learning against poisoning attacks through local update amplification," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 1241–1250, 2024.

[5] C. Dong, J. Weng, M. Li, J.-N. Liu, Z. Liu, Y. Cheng, and S. Yu, "Privacy-preserving and byzantine-robust federated learning," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–16, 2023.

[6] Z. Alebouyeh and A. J. Bidgoly, "Benchmarking robustness and privacy-preserving methods in federated learning," *Future Generation Computer Systems*, vol. 155, pp. 18–38, 2024.

[7] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "SEAR: Secure and efficient aggregation for byzantine-robust federated learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 5, pp. 3329–3342, 2022.

[8] E. M. El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[9] J. Shi, W. Wan, S. Hu, J. Lu, and L. Yu Zhang, "Challenges and approaches for mitigating byzantine attacks in federated learning," in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2022.

[10] S. Li, E. C.-H. Ngai, and T. Voigt, "An experimental study of byzantine-robust aggregation schemes in federated learning," *IEEE Transactions on Big Data*, pp. 1–13, 2023.

[11] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, and Y. Liu, "LoMar: A local defense against poisoning attack on federated learning," 2023.

[12] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[13] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[14] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020.

[15] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022.

[16] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, 2019.

[17] S. P. Karimireddy, L. He, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via bucketing," in *International Conference on Learning Representations (ICLR)*, 2022.

[18] K. Kumari, P. Rieger, H. Fereidooni, M. Jadliwala, and A.-R. Sadeghi, "BayBFed: Bayesian backdoor defense for federated learning," in *IEEE Symposium on Security and Privacy (SP)*, 2023.

[19] K. Hsieh, A. Phanishayee, O. Mutlu, and P. Gibbons, "The non-iid data quagmire of decentralized machine learning," in *International Conference on Machine Learning (ICML)*, 2020.

[20] Z. Li, Y. Sun, J. Shao, Y. Mao, J. H. Wang, and J. Zhang, "Feature matching data synthesis for non-IID federated learning," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024.

[21] J. Xue, M. Liu, S. Sun, Y. Wang, H. Jiang, and X. Jiang, "Fedbiad: Communication-efficient and accuracy-guaranteed federated learning with bayesian inference-based adaptive dropout," in *IEEE International Parallel and Distributed Processing Symposium*, 2023.

[22] G. Shirvani, S. Ghasemshirazi, and B. Beigzadeh, "Federated learning: Attacks, defenses, opportunities, and challenges," *arXiv preprint arXiv:2403.06067*, 2024.

[23] N. Rodríguez-Barroso, E. Martínez-Cámara, M. V. Luzón, and F. Herrera, "Dynamic defense against byzantine poisoning attacks in federated learning," *Future Generation Computer Systems*, vol. 133, pp. 1–9, 2022.

[24] Q. Xu, Z. Yang, Y. Zhao, X. Cao, and Q. Huang, "Rethinking label flipping attack: From sample masking to sample thresholding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7668–7685, 2023.

[25] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," in *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, 2012.

[26] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," 2021.

[27] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *Uncertainty in Artificial Intelligence*, 2020.

[28] Y. Liu, C. Chen, L. Lyu, F. Wu, S. Wu, and G. Chen, "Byzantine-robust learning on heterogeneous data via gradient splitting," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[29] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLtrust: Byzantine-robust federated learning via trust bootstrapping," in *Network and Distributed Systems Security Symposium (NDSS)*, 2021.

[30] Y. Li, X. Lyu, X. Ma, N. Koren, L. Lyu, B. Li, and Y.-G. Jiang, "Reconstructive neuron pruning for backdoor defense," in *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[31] Y. Yang, K. Wang, and W. Li, "FedREP: A byzantine-robust, communication-efficient and privacy-preserving framework for federated learning," *arXiv preprint arXiv:2303.05206*, 2023.

[32] M. H. Meng, S. G. Teo, G. Bai, K. Wang, and J. S. Dong, "Enhancing federated learning robustness using data-agnostic model pruning," *Advances in Knowledge Discovery and Data Mining*, 2023.

[33] X. Zhang, Y. Li, W. Li, K. Guo, and Y. Shao, "Personalized federated learning via variational Bayesian inference," in *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[34] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning (ICML)*, 2016.

[35] E. B. A. Chérief, "Convergence rates of variational inference in sparse deep learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[36] Y. Wang and D. M. Blei, "Frequentist consistency of variational bayes," *arXiv preprint arXiv:1705.03439*, 2018.

[37] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[38] P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. Wilson, "What are bayesian neural network posteriors really like?," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

[39] M. Magris and A. Iosifidis, "Bayesian learning for neural networks: an algorithmic survey," *arXiv preprint arXiv:2211.11865*, 2023.

[40] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[41] opensignal.com., "USA mobile network experience report January 2024." https://www.opensignal.com/reports/2024/01/usa/mobile-network-experience.

[42] P. Alquier and J. Ridgway, "Concentration of tempered posteriors and of their variational approximations," *arXiv preprint arXiv:1706.09293*, 2019.

[43] A. Garriga-Alonso and V. Fortuin, "Exact langevin dynamics with stochastic gradients," in *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

[44] Y. Gal, "Uncertainty in deep learning," *PhD thesis, University of Cambridge*, 2016.

[45] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-Pruning: Defending against backdooring attacks on deep neural networks,"

[46] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[47] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *International joint conference on neural networks (IJCNN)*, 2019.

[48] T. Chu, A. Garcia-Recuero, C. Iordanou, G. Smaragdakis, and N. Laoutaris, "Securing federated sensitive topic classification against poisoning attacks," in *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2023.

[49] N. G. Polson and V. Ročková, "Posterior concentration for sparse deep learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[50] R. Nakada and M. Imaizumi, "Adaptive approximation and generalization of deep neural network with intrinsic dimensionality," *The Journal of Machine Learning Research*, vol. 21, no. 1, 2020.

[51] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022.

[52] T. Huang, S. Liu, L. Shen, F. He, W. Lin, and D. Tao, "Achieving personalized federated learning with sparse local models," *arXiv preprint arXiv:2201.11380*, 2022.

[53] R. Dai, L. Shen, F. He, X. Tian, and D. Tao, "DisPFL: Towards communication-efficient personalized federated learning via decentralized sparse training," in *International Conference on Machine Learning (ICML)*, 2022.

[54] X. Jiang and C. Borcea, "Complement sparsification: Low-overhead model pruning for federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[55] Z. Jiang, Y. Xu, H. Xu, Z. Wang, C. Qiao, and Y. Zhao, "FedMP: Federated learning through adaptive model pruning in heterogeneous edge computing," in *IEEE 38th International Conference on Data Engineering (ICDE)*, 2022.

[56] S. Caldas, J. Konečny, H. B. McMahan, and A. Talwalkar, "Expanding the reach of federated learning by reducing client resource requirements," *arXiv preprint arXiv:1812.07210*, 2018.

[57] D. Wen, K. J. Jeon, and K. Huang, "Federated dropout – a simple approach for enabling federated learning on resource constrained devices," *IEEE Wireless Communications Letters*, vol. 11, pp. 923–927, 2022.

[58] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010.

[59] X. Han, R. Kashif, and V. Roland, "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[60] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[61] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, B. McMahan, V. Smith, and A. Talwalkar, "LEAF: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2019.

[62] A. Li, J. Sun, B. Wang, L. Duan, S. Li, Y. Chen, and H. Li, "LotteryFL: Empower edge intelligence with personalized and communication-efficient federated learning," in *IEEE/ACM Symposium on Edge Computing (SEC)*, 2021.

[63] S. Bibikar, H. Vikalo, Z. Wang, and X. Chen, "Federated dynamic sparse training: Computing less, communicating less, yet learning better," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[64] Y. Miao, X. Yan, X. Li, S. Xu, X. Liu, H. Li, and R. H. Deng, "RFed: Robustness-enhanced privacy-preserving federated learning against poisoning attack," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5814–5827, 2024.

[65] Z. Lu, S. Lu, Y. Cui, X. Tang, and J. Wu, "Split aggregation: Lightweight privacy-preserving federated learning resistant to byzantine attacks," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 5575–5590, 2024.

[66] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[67] M. Wu, B. Zhao, Y. Xiao, C. Deng, Y. Liu, and X. Liu, "MODEL: A model poisoning defense framework for federated learning via truth discovery," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 8747–8759, 2024.

[68] Z. Chen, S. Yu, M. Fan, X. Liu, and R. H. Deng, "Privacy-enhancing and robust backdoor defense for federated learning on heterogeneous data," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 693–707, 2024.

[69] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[70] J. Wolfrath, N. Sreekumar, D. Kumar, Y. Wang, and A. Chandra, "HACCS: Heterogeneity-aware clustered client selection for accelerated federated learning," in *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2022.

[71] N. Bouacida, J. Hou, H. Zang, and X. Liu, "Adaptive federated dropout: Improving communication efficiency and generalization for federated learning," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021.