Fed-MS: Fault Tolerant Federated Edge Learning with Multiple Byzantine Servers

Senmao Qi[1], Hao Ma[1], Yifei Zou[1*], Yuan Yuan[1], Peng Li[2], and Dongxiao Yu[1]
[1]School of Computer Science and Technology, Shandong University, Qingdao, China
[2]School of Computer Science and Technology, University of Aizu, Fukushima, Japan

2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00095

***Abstract*—Due to its decentralized framework and outdoor environments, federated edge learning (FEEL) faces significant vulnerability to malicious attacks within edge networks. Prevailing FEEL approaches typically hinge on a dependable parameter server (PS) to contend with the adversarial updates from Byzantine clients. Recognizing the inherent unreliability of PSs in edge networks, this paper delves into the security challenges of FEEL, specifically addressing Byzantine PSs. We present a Byzantine fault-tolerant FEEL algorithm, named Fed-MS, in which a multi-server technique along with a newly designed trimmed-mean-based model filter is employed. This combination ensures that each client can obtain a feasible global model for its local training, closely approximating a true model aggregated by benign PSs. Furthermore, we propose a sparse uploading strategy in Fed-MS to enhance communication efficiency for model aggregation to multiple PSs. Theoretical analysis demonstrates that, when Byzantine PSs are a minority, Fed-MS achieves an expected convergence speed of $\mathcal{O}(1/T)$ with T defined as the number of training rounds, akin to state-of-the-art works under non-Byzantine settings. Extensive experiments are conducted on the CIFAR-10 dataset with MobileNet V2 as the training model. The numerical results show that our Fed-MS can improve the model accuracy from 10% to at least 76% under the malicious attacks from Byzantine PSs. Our code is released at https://github.com/haoma2772/Fed-MS.**

***Index Terms*—Federated Learning, Edge Networks, Byzantine Fault Tolerance.**

I. Introduction

As an interdisciplinary of artificial intelligence and networking, the federated edge learning (FEEL) enables large-scale and privacy-preserving machine learning (ML) on the edge scenarios and has witnessed significant growth in Industrial Internet of Things networks [1], [2]. Under the federated learning (FL) framework, multiple clients can obtain a high-quality global ML model by only sharing their local ML models instead of data to others, which protects the privacy of users. Besides, the edge-based parameter server (PS) provides the clients with fast and green model exchange. Thus, the FEEL has a higher efficiency and lower communication overhead, compared with the traditional cloud-based FL framework [3].

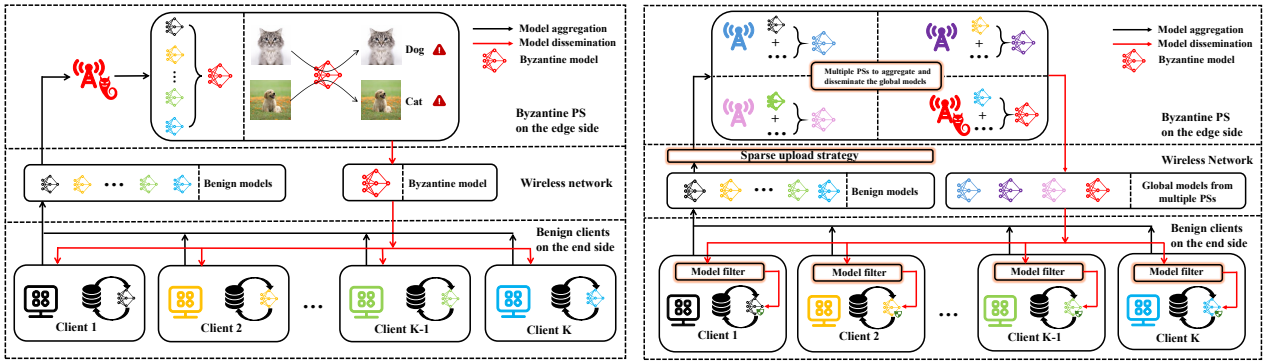
Despite the advantages offered by the combination of FL and edge computing, the openness and outdoor environments of edge networks have posed new challenges on the reliability and security of FEEL. Specifically, on the end side, the openness of edge networks allows the adversarial clients to easily join the FL process and degrade the performance of the global model by uploading malicious local updates [4], [5]. Meanwhile, on the edge side, most the edge-based PSs are deployed in some outdoor environments and they are more susceptible to attacks compared to those in well-protected data centers [3]. Since most of the existing FL schemes rely on a reliable PS to aggregate and disseminate the global ML model, they would fail if the outdoor PS is attacked and controlled by an adversary.

To address the above security problems, a series of Byzantine fault tolerant strategies have been proposed to enhance the reliability and security of federated learning process, based on statistics, clustering, and differential privacy techniques. In statistics-based strategies, the median, geometric median and trimmed mean statistics are adopted in [6], [7] to find a representative estimation of true results by removing the suspiciously large or small parameters in each dimension of the uploaded local model, which can mitigate the negative impacts from the malicious clients. Besides, an iterative Weiszfeld algorithm is employed in FL via over-the-air computation, effectively computing the smoothed geometric median aggregation result to counter Byzantine attack [8]. Additionally, the cluster-based methods, such as Krum [9], Bulyan [10], AFA [11], Fools-Gold [12], and Auror [13], have been designed to identify and remove Byzantine models or gradients. In recent years, the differential privacy technique has been used to mitigate the impact of backdoor attacks from Byzantine clients by adding random noise to the pruned model or gradient [14], [15]. Even though the aforementioned works can resist Byzantine attacks from the malicious clients on the end side, all of them rely on a reliable PS to realize the global model aggregation and dissemination. To the best of our knowledge, few of the existing works consider the FEEL problem with unreliable or even Byzantine PSs.

In this paper, we consider the federated edge learning problem with the Byzantine edge-based PSs. Unlike the PSs located in the well-protected data centers, many edge-based PSs are deployed in an outdoor environment and faces various attacks from the adversaries [16]. If a PS is attacked and controlled by an adversary, the aggregated global model can be arbitrarily tampered and delivered to the clients. Then, the FL process moves to an unconvergent direction even though all the clients are the benign ones. In Fig. 1(a), we show how a Byzantine PS mislead the benign clients with a malicious

* The corresponding author is Yifei Zou (yfzou@sdu.edu.cn).

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00095



(a) The existing FL algorithms is hard to survive if its PS is Byzantine.

(b) Our Fed-MS is Byzantine resilient with Multiple PSs.

Fig. 1: The existing works rely on single PS to aggregate and disseminate the global model. If the PS is a Byzantine one, it can mislead the benign clients with a malicious global model; Our FedMS has multiple PSs to participate in the model aggregation and dissemination, which attenuate the negative impacts from Byzantine PSs.

global model. Since most of the existing Byzantine-resilient FL strategies rely on a reliable PS to aggregate the non-Byzantine global model, they no longer have the provable performance if the PS was attacked. Thus, we can believe that the investigation on the FEEL with unreliable PSs in this paper is necessary and significant.

To defend against the potential Byzantine PSs on the edge side, a Byzantine-resilient **Fed**erated edge learning algorithm with **M**ultiple **S**ervers (termed as **Fed-MS** for short) is proposed in this paper. As is illustrated in the Fig. 1(b), our Fed-MS comprises multiple PSs on the edge side and a group of clients on the end side. We say a PS is a Byzantine one if it is attacked and controlled by an adversary. A Byzantine PS can deviate arbitrarily from the protocol they are specified to execute to cause a wide variety of faults. Otherwise, we say the PS is a benign one. We assume that the distribution of the Byzantine PSs on the edge side is unknown for the clients and can be arbitrary, but the total number of Byzantine PSs should be the minority compared with the number of benign PSs. Otherwise, the FEEL problem with Byzantine PSs is unfeasible. Considering that the clients have little knowledge on which PSs are the Byzantine ones, the clients in our Fed-MS upload their local models to multiple PSs. The multiple PSs aggregate the uploaded local models and deliver the aggregated results to the clients in parallel. Since the Byzantine PSs are the minority, the majority of the aggregated global models received by a client are benign. To figure out a non-Byzantine global model from those benign ones, a trimmed-mean-based model filter is designed for each client, the input of which are those Byzantine and benign global models and the output of which is a feasible global model that is not far away from the benign global models. Such a feasible global model selected by the trimmed-mean-based model filter will be used by clients in the next-round local training. Additionally, considering that uploading the local models to the multiple PSs overloads the resource-constrained edge network, a sparse uploading strategy is designed for each client to guarantee the communication efficiency of our Fed-MS algorithm. With the sparse uploading strategy, our Fed-MS has the same communication overhead with a general FL process based on single PS on model aggregation. Finally, we prove that our Fed-MS achieves a convergence speed of $\mathcal{O}(1/T)$ in expectation with T defined as the number of training rounds, which is similar with the results in the state-of-the-art [17], [18], and [19] under non-Byzantine setting. Extensive simulations are conducted to verify the performance of our algorithm. The contributions of our work are listed in the following:

- Different from most of the previous FL works that rely on reliable PSs to address the Byzantine behaviors from the clients, this paper is the first one considering the FL with Byzantine PSs, which is a realistic and significant security problem when FL is extended to the outdoor edge computing environments.
- To address the FL problem with Byzantine PSs, we propose a Byzantine-resilient **Fed**erated edge learning algorithm with **M**ultiple **S**ervers, termed as **Fed-MS** for short. To defend the malicious global models aggregated by the Byzantine PSs, multiple PSs are used in Fed-MS for global model aggregation and dissemination. A trimmed mean-based model filter is adopted by each clients to figure out a feasible global model for next-round local training, when it receives multiple global models from the PSs. Besides, a sparse uploading strategy is designed in Fed-MS for the communication efficient model aggregation. With the sparse uploading strategy, the communication overload of our Fed-MS on model aggregation is the same with those classical FL works with single PS.
- With theoretical proofs, we show that when the number of Byzantine PSs is the minority, our Fed-MS algorithm achieves the convergence speed of $\mathcal{O}(1/T)$ with T defined as the number of training rounds. Our result is the same with that in the state-of-the-art works [17]–[19]

under non-Byzantine setting.
- Extensive experiments are conducted on the CIFAR-10 dataset with MobileNet V2 proposed in [20]. Four common Byzantine attacks including Noise, Random, Safeguard and Backward attacks [21] are deployed on the PSs. Numerical results show that our Fed-MS can achieve about $70\% \sim 76\%$ prediction accuracy after 60 rounds of training. Compared with 10% accuracy of the vanilla FedAvg, Fed-MS has a strong Byzantine resilience.

Roadmap. The rest of this paper is organized as follows. We present the related work in Section II. The FEEL system, Byzantine model and problem formulation are given in Section III. Our Fed-MS algorithm is introduced in Section IV, with its convergence proved in Section V and numerical results reported in Section VI. In Section VII, we conclude our paper.

II. Related Work

Byzantine Fault Tolerant Federated Learning. The distributed framework of FL allows the Byzantine clients to share malicious updates without being detected, which reduces the accuracy of the global model. In recent years, several Byzantine fault tolerant methods have been proposed to mitigate the disruption caused by the Byzantine updates from the clients, such as the statistics-based strategies in [6]–[8], [22]–[24], the clustering methods in [9]–[13] and the differential privacy techniques in [14], [15], [25]. Specifically, robust statistics such as median and trimmed mean are used in [6] to help the PS find a probable estimation of the multiple models or gradients updated from the clients, some of which may be Byzantine. By calculating the median or trimmed mean of each dimension of the received parameter results, the PS in [22] removes the impact of extreme values on the aggregation results and obtains an error-bounded estimation on the true results. Similar statistics also include geometric medians in [7] and [8], which are widely used to defense the Byzantine attacks from the clients. In addition, Krum in [9], Bulyan in [10], AFA in [11], Fools-Gold in [12] and Auror in [13] use the clustering schemes to identify and remove the malicious results uploaded from Byzantine clients. The clustering-based studies assume that the local data of benign clients follows certain specific distributions, such as i.i.d.. So the models or gradients uploaded by benign clients are similar with each other, and different from the results uploaded by malicious clients. In recent years, the differential privacy model aggregation has been proven to be an efficient method to defend against the differential attacks and gradient inversion attacks from Byzantine clients [14], [15], [25]. Whereas, most of the existing Byzantine-resilient works consider the potential attacks from the malicious clients and rely on a reliable PS to implement their defense strategies. In this paper, we consider the relative Byzantine resilient problem with unreliable PSs, which is a realistic security problem when the FL are extended to the edge computing environment and harder than the previous Byzantine-resilient works relying on a reliable PS.

Federated Learning with Multi-Servers. In this paper, the multi-servers are used in our FEEL algorithm to defend against the edge-side Byzantine attacks. Honestly speaking, the multi-server technique has already been used to solve the heterogeneous, personalized, and split federated learning problems. For instance, the clients with similar capabilities or data distribution will be grouped in hierarchical FL. Within each group, the model aggregation is performed using a dedicated parameter server to mitigate the challenges arising from the heterogeneity or personality among various clients [26]–[29]. In [30], the convergence analysis under the hierarchical federated learning with multiple PSs is given. In [31], [32], the main server and Fed server are used in split federated learning for model update and aggregation respectively. To the best of our knowledge, few of the works address the Byzantine fault tolerant FL problem with multi-server technique.

III. System Model and Problem Definitions

In this section, our FEEL system model, the Byzantine model for edge-based PSs and the problem definition for Byzantine resilient FEEL problem are introduced one by one.

A. Federated Edge Learning Model with Byzantine PSs

We consider a FEEL system with K clients on the end side for local training and P parameter servers on the edge side for global model aggregation and dissemination. Each client k has a local training dataset D_k and a local objective function $F_k(w; D_k) \triangleq \frac{1}{|D_k|}\sum_{\{x,y\}\in D_k}\mathcal{L}(w;\{x,y\})$, where w is the model parameter of d dimensions and $\mathcal{L}(w;\{x,y\})$ is the corresponding loss function on a specific sample $\{x,y\}$. The goal of our FEEL is to find a optimal global model parameters w^* that can minimize the linear combination of each local object function:

$$w^* = \arg\min_{w\in\mathbb{R}^d} \frac{1}{K}\sum_{k=0}^{K-1} F_k(w; D_k) \tag{1}$$

We use the classical synchronized method of FedAvg [33] to optimize the goal in (1), which includes the following three stages in each training round t.

- **Local Training.** Each client k conducts mini-batch Stochastic Gradient Descent (SGD) for E times to optimize the local objective $F_k(w; D_k)$, i.e. $w^k_{t,i+1} = w^k_{t,i} - \eta_{t,i}\nabla F_k(w^k_{t,i}, \xi^k_{t,i})$, for $i = 0, 1, ..., E-1$. $w^k_{t,i}$ denotes i-th local model updates of client k in the training round t, $\nabla F_k(w^k_{t,i}, \xi^k_{t,i})$ is the gradient of the given parameters, $\eta_{t,i}$ is the corresponding learning rate and $\xi^k_{t,i}$ is a mini-batch randomly sampled from the local dataset D_k.
- **Model Aggregation.** Each client k selects one or multiple PSs to upload its local model parameters $w^k_{t,E}$. For each PS i, it averages the received local models from the clients, i.e $a^i_{t+1} = \frac{1}{|N_i|}\sum_{k\in N_i} w^k_{t,E}$, where N_i is the set of clients that select PS i to upload its local model.
- **Model Dissemination.** Each PS i disseminates its aggregated result a^i_{t+1} to the clients. After receiving multiple global models from all the PSs, each client has to figure out a feasible model to start its next-round local training.

The above process will continue until the ML model on each client converges.

Capability and Knowledge of Byzantine PSs. We assume that some parameter servers have Byzantine behavior, that is, they can produce some misleading results to clients, degrading the efficiency of federated learning. In this paper, we consider a strong Byzantine behavior whose characteristics are summarized as follows:

- **Unknown Distribution.** We assume that there are B Byzantine PSs hidden behind all the P PSs. The distribution of the Byzantine PSs can be arbitrary and unknown for the clients. The only restriction is $B \leq P/2$, which indict Byzantine PSs is minority.
- **Arbitrary Tampering.** After aggregating the local models from multiple clients, the Byzantine PS can arbitrarily tamper the aggregated global model in its model dissemination step. In a worst case, a Byzantine PS can send various tampered models to different clients. Such a Byzantine behavior cannot be detected since the clients cannot directly communicate with each other.
- **Adaptive Knowledge.** The Byzantine PSs can have a full knowledge on the FEEL algorithm, the history and current state of the FL process, and can adapt their behaviors according to the obtained information. Such a setting is termed as adaptive adversary in [34].

Capability and Knowledge of Clients. We assume that all clients have the same capabilities on their training devices and are synchronized in the above local training, model aggregation, and model dissemination stages. Clients know the total number of edge-based PSs P but has little knowledge for the Byzantine PSs, except their number B. When receiving multiple global models from the PSs, it is hard for the clients to distinguish whether a global model is tampered by a Byzantine PS. Important notations in the model, algorithm description and analysis sections are summarized in Table I.

Notations	Definition
$K, \mathcal{K}$	Number, set of clients
$P, \mathcal{P}$	Number, set of PSs
$B, \mathcal{B}$	Number, set of Byzantine PSs
D_k	Local training data of client k
$F_k(\cdot)$	Local object of client k
$\nabla F_k(\cdot)$	Gradient of F_k
$\xi_{t,i}^k$	A mini-batch of client k
$\mathcal{L}(\cdot,\cdot)$	Loss function
$w_{t,i}^k$	i-th local model parameter of client k in round t
E	Number of local iterations
w^*	Optimal global model
$\|\cdot\|_2$	L_2 norm
$\eta_{t,i}$	Learning rate of i-th local iteration in round t
a_t^i	Aggregation result of PS i in round t
$\tilde{a}_t^i$	Dissemination result from PS i in round t
N_i	Set of clients that select PS i
β	Trimmed rate

TABLE I: Important notations.

B. Problem Definition

To overcome the potential attacks from the Byzantine PSs, when a client receives multiple aggregated models from the PSs, it has to figure out a feasible global model that is not far away from the global models aggregated by the benign PSs. Therefore, the key component in the Byzantine fault tolerant FL problem with Byzantine PSs is to find a defense method $\mathbf{Def}(\ )$, the input of which are those Byzantine and benign global models and the output of which is a feasible global model for the next-round local training. With the help of $\mathbf{Def}(\ )$, the clients finally obtain the high-performance ML models. Let $\tilde{a}_t^i$ denote the dissemination result of PS i in round t, which is equal to a_t^i for a benign PS and can be arbitrary for a Byzantine PS. Then, our objective can be expressed in the following:

$$\begin{aligned} &\min_{w\in\mathbb{R}^d} \frac{1}{K}\sum_{k=0}^{K-1} F_k(w; D_k), \\ \text{with } (1)\ & w_{t+1,0}^k = \mathbf{Def}(\tilde{a}_t^0, \tilde{a}_t^1, ..., \tilde{a}_t^{P-1}),\ k\in\mathcal{K},\ t=0,1,2,... \\ (2)\ & w_{t,i+1}^k = w_{t,i}^k - \eta_{t,i}\nabla F_k(w_{t,i}^k, \xi_{t,i}^k), \\ & \qquad k\in\mathcal{K},\ i=0,1,...,E-1,\ t=0,1,2,... \\ (3)\ & a_{t+1}^i = \frac{1}{|N_i|}\sum_{k\in N_i} w_{t,E}^k,\ i\in\mathcal{P},\ t=0,1,2,... \end{aligned} \tag{2}$$

In the above objective, $\mathcal{K}, \mathcal{P}$ and $\mathcal{B}$ represent the set of clients, parameter servers and Byzantine nodes respectively. Our goal is to find a feasible global model from mixed bag aggregation results from multiple servers by using $\mathbf{Def}(\ )$. In this paper, we use statistics-based trimmed mean to achieve this goal, and provide the corresponding theoretical analysis.

IV. Byzantine-resilient FEEL Algorithm

A. Challenges and Solutions in Algorithm Description.

We consider the Byzantine resilient FEEL problem with multiple PSs, some of which are Byzantine. Intuitively speaking, if the distribution of Byzantine PSs was known by the clients, the FEEL problem can be solved by letting all the clients choose the benign PSs to do the model aggregation and dissemination. Considering that the clients have little knowledge of Byzantine PSs, we have the following two challenges in our algorithm design. The first challenge is how to aggregate sufficient local models to the benign PSs, especially when the benign PSs cannot be distinguished by the clients. The second challenge is when a client receives multiple global models from the Byzantine and Benign PSs, how can it figure out a feasible global model for next-round local training. We say a global model is feasible when it is not far away from the global models aggregated by the benign PSs. Only when the first challenge is solved, can we guarantee that there are sufficient local models aggregated by the benign PSs. When the second challenge is solved, the clients can make full use of the global models provided by the benign PSs, which speeds up their local convergence processes.

To overcome the first challenge, a trivial approach is to let each client upload its local model to all the PSs. Thus, each of the benign clients must receive the local models from all the clients. Whereas, the communication cost would be $K \times P$, which is P times larger than that in a classical FL with single PS. Considering that such a communication cost may overload

the resource-constrained edge network, a sparse uploading strategy is adopted in our algorithm for a communication efficient model uploading. In our sparse uploading strategy, each client randomly and uniformly choose a PS to upload its local model. By doing this, the communication cost is K, which is the same with that in a classical FL with single PS. The trade-off of our strategy is that each client receives the various local model from the different clients. Thus, their aggregated results are not the same, which makes our proofs on convergence harder.

To address the second challenge, a trimmed-mean-based model filter is design for each client. Specifically, when a client receives multiple global models from the Byzantine and benign PSs, a fraction of largest and smallest parameters in each dimension of the global model will be removed. Then, the remaining parameters in each dimension will be averaged by the client to form a feasible global model for its next-round local training. Since a Byzantine PS can disseminate inconsistent models to different client, the feasible global models obtained by different clients are not the same, which also makes our proofs on convergence harder.

B. Detailed Description for Algorithm Design

Similar with the classical FedAvg algorithm [17], our Byzantine fault tolerant Fed-MS algorithm also consists of three stages in each training round t. In the first local training stage, each client k performs E local iterations, where each iteration utilizes a mini-batch sampled from its local dataset D_k for stochastic gradient descent. In formal words, $w_{t+i+1}^k \leftarrow w_{t+i}^k - \eta_{t+i}\nabla F_k(w_{t+i}^k)$.

Then, we proceed to the model aggregation stage. Note that there are P PSs on the edge side, B of which are the Byzantine ones. Each client k randomly and uniformly selects a PS from the P PSs to upload its latest local model w_{t+E}^k. Meanwhile, each of the benign PSs averages the received local models to obtain its global model. However, the Byzantine PSs can perform arbitrary in model aggregation stage.

In the model dissemination stage, the benign PSs disseminate its global model obtained in aggregation stage to all the clients. For the Byzantine PSs, it can arbitrarily tamper its aggregated global model and disseminate the tampered global model to the clients. After receiving the global models from the P PSs, each client computes the trimmed mean $\mathbf{trmean}_\beta\{\ \}$ for all results where $\beta = B/P$ is the trimmed rate. Specifically, in each dimension, the parameters of the largest and smallest β components are discarded and the remaining results are averaged. The result computed by trimmed mean method will be regarded as a feasible global model and used in the next-round local training. The pseudocode of our Fed-MS is given in Algorithm 1.

An example for the trimmed mean function. For example, $\mathbf{trmean}_{0.2}\{1, 2, 3, 4, 5\}$ will remove 20% of the smallest and largest values, i.e. 1 and 5, and then average the remaining results, i.e. $(2 + 3 + 4)/3 = 3$.

Algorithm 1: Byzantine resilient Fed-MS

For each parameter server i:

for *each round* $t = 0, 1, 2, ...$ **do**
  Wait for clients to finish local training;
  `// Local training stage`
  Receive the local models from the clients in set N_i;
  `// $N_i$ is the set of clients that upload local models to PS $i$`
  $a_{t+1}^i = \frac{1}{|N_i|}\sum_{k\in N_i} w_{k,t,E}$;
  `// Model aggregation stage`
  Broadcast a_{t+1}^i to all the clients;
  `// Model dissemination stage`

For each client k:,

Initialization: $w_{k,0,0} = w_0$;
for *each round* $t = 0, 1, 2, ...$ **do**
  for *each epoch* $i = 0, 1, ..., E-1$ **do**
    Randomly sample a mini-batch $\xi_{t,i}^k$ from D_k;
    $w_{t,i+1}^k = w_{t,i}^k - \eta_{t,i}\nabla F_k(w_{t,i}^k, \xi_{t,i}^k)$;
  `// Local training stage`
  Randomly select a PS to upload $w_{k,t,E}$;
  `// Model aggregation stage`
  Receive the global models from P parameter servers, denoted by $\tilde{a}_{t+1}^0, \tilde{a}_{t+1}^1, ..., \tilde{a}_{t+1}^{P-1}$;
  $w_{k,t+1,0} = \mathbf{trmean}_\beta\{\tilde{a}_{t+1}^0, \tilde{a}_{t+1}^1, ..., \tilde{a}_{t+1}^{P-1}\}$;
  `// Model dissemination stage`

V. Convergence Analysis

A. Notations and Assumptions

Considering that each client will perform E local iterations in each round, for the convenience, we borrow the notations used in [17] and [35] to analyze the impact of each step of mini-batch SGD on model update. Specifically, we introduce two additional variable v_t^k, e_t^k to rewrite the iterative process of Byzantine fault tolerant FL into the following form.

$$\begin{aligned} v_{t+1}^k &= w_t^k - \eta_t \nabla F_k(w_t^k, \xi_t^k) \\ w_{t+1}^k &= \begin{cases} v_{t+1}^k & \text{, if } t+1 \notin I_E \\ e_{t+1}^k & \text{, if } t+1 \in I_E \end{cases} \end{aligned} \tag{3}$$

In the above equation, v_{t+1}^k is used to reflect the direct one step model parameters after a mini-batch SGD. $I_E = \{nE | n \in \mathbb{N}\}$ represents all the steps where we conduct model aggregation. e_{t+1}^k represents the feasible global model of client k at step $t+1$ by using trimmed mean, which is defined as:

$$\begin{aligned} &e_{t+1}^k = \mathbf{trmean}_\beta\{\tilde{a}_{t+1}^0, \tilde{a}_{t+1}^1, ..., \tilde{a}_{t+1}^{P-1}\}, \\ &\text{s.t. } \frac{1}{|N_i|} a_{t+1}^i = \sum_{k\in N_i} v_{t+1}^k, \forall i \in \mathcal{P} \\ &\qquad \tilde{a}_{t+1}^i = a_{t+1}^j, \tilde{a}_{t+1}^j = \text{Any Value}, i \in \mathcal{P} \setminus \mathcal{B}, \forall j \in \mathcal{B} \end{aligned} \tag{4}$$

The above equation represents the result of multiple PSs aggregation and the process of global model estimation based on trimmed mean. We define two virtual sequences $\overline{v}_t =$

$\frac{1}{K}\sum_{k=0}^{K} v_t^k$ and $\overline{w}_t = \frac{1}{K}\sum_{k=0}^{K} w_t^k$. Similarly, we additionally introduce $\overline{a}_t = \frac{1}{K}\sum_{k=0}^{K} a_t^i$ and $\overline{e}_t = \frac{1}{K}\sum_{k=0}^{K} e_t^k$. At last, we define $g_t = \frac{1}{K}\sum_{k=0}^{K} \nabla F_k(w_t^k, \xi_t^k)$ and $\overline{g}_t = \mathbb{E}(g_t) = \frac{1}{K}\sum_{k=0}^{K} \nabla F_k(w_t^k)$. The above newly introduced variables reflect the average value of the corresponding variable, making it convenient for us to bound their variances.

Compared with the classical FedAvg in which the local models of the all clients contribute to a global model, the error in our Fed-MS mainly comes from two parts: the first part is under sparse uploading, thus there is a certain error between $\overline{a}_t$ and $\overline{v}_t$ before. The second part is due to the existence of Byzantium, $\overline{e}_t$, as an estimation of $\overline{a}_t$, is also biased. This in turn leads to differences in the expectations of $\overline{w}_t$ and $\overline{v}_t$. Noting that in the absence of Byzantine nodes, the expectations of them are the same. Before formally starting the convergence analysis, we make the following assumptions on the local object function:

Assumption 1. *For any client k, F_k is L-smooth, i.e., for all $\boldsymbol{u}$ and $\boldsymbol{v}$, $F_k(\boldsymbol{u}) \leq F_k(\boldsymbol{v}) + (\boldsymbol{u}-\boldsymbol{v})^T \nabla F_k(\boldsymbol{v}) + \frac{L}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_2^2$.*

Assumption 2. *For any client k, F_k is μ-strongly convex, i.e., for all $\boldsymbol{u}$ and $\boldsymbol{v}$, $F_k(\boldsymbol{u}) \geq F_k(\boldsymbol{v}) + (\boldsymbol{u}-\boldsymbol{v})^T \nabla F_k(\boldsymbol{v}) + \frac{\mu}{2}\|\boldsymbol{u}-\boldsymbol{v}\|_2^2$.*

Assumption 3. *For any client k, let ξ_t^k be a mini-batch sampled from local dataset D_k uniformly and randomly. The variance of stochastic gradient is bounded, i.e., $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k) - \nabla F_k(w_t^k)\|_2^2 \leq \sigma_k^2$, where $\nabla F_k(w_t^k)$ is the expectation of $F_k(w_t^k, \xi_t^k)$.*

Assumption 4. *For any client k, the expected expected squared norm of stochastic gradients is uniformly bounded, i.e., $\mathbb{E}\|\nabla F_k(w_t^k, \xi_t^k)\|_2^2 \leq G^2$ for all time-step t.*

It is worth mentioning that the assumption 1-4 are very common assumptions in theoretical federated learning [17]–[19], [22]. Here we give conclusions about the convergence of the algorithm.

B. Analysis for estimation error

According to the variables defined above, we can easily obtain the following relationship:

$$\begin{aligned} \overline{v}_{t+1} &= \overline{w}_t - \eta_t * g_t \\ \overline{w}_{t+1} &= \begin{cases} \overline{v}_{t+1} & \text{, if } t+1 \notin I_E \\ \overline{e}_{t+1} & \text{, if } t+1 \in I_E \end{cases} \end{aligned} \tag{5}$$

It can be found that when $t \in I_E$, $\overline{w}_{t+1}$ and $\overline{v}_{t+1}$ are not equal, and $\overline{w}_{t+1}$ is a biased estimate of $\overline{v}_{t+1}$, so we first analyze the error of this estimation.

Lemma 1. *If assumption 4 holds and η_t is non-increasing with $\eta_t \leq 2\eta_{t+E}$, we have:*

$$\mathbb{E}(\frac{1}{K}\sum_{k=0}^{K-1} \|\overline{w}_t - w_t^k\|_2^2) \leq 4\eta_t^2 E^2 G^2$$

Proof. Consider an interval $I = [t_0, t_0 + E]$ where $t_0 = nE$ and $n \in \mathbb{N}$, for any $t \in I$, we have:

$$\begin{aligned} &\mathbb{E}(\frac{1}{K}\sum_{k=0}^{K-1} \|\overline{w}_t - w_t^k\|_2^2) \\ =&\mathbb{E}(\frac{1}{K}\sum_{k=0}^{K-1}\sum_{k=0}^{K-1} \|w_t^k - \overline{w}_{t_0} + \overline{w}_{t_0} - w_t^k\|_2^2) \\ \leq&\mathbb{E}\sum_{k=0}^{K-1} \frac{1}{K}\|w_t^k - \overline{w}_{t_0}\|_2^2 \leq \mathbb{E}\sum_{k=0}^{K-1} \frac{1}{K}\|w_t^k - w_{t-1}^k + ... - \overline{w}_{t_0}\|_2^2 \\ \leq&\sum_{k=0}^{K-1} \frac{1}{K}\sum_{t=t_0+1}^{t} (t-t_0)\eta_{t_0}^2 G^2 \leq \sum_{k=0}^{K-1} \frac{\eta_{t_0}^2}{K} E^2 G^2 \leq 4\eta_t^2 E^2 G^2 \end{aligned} \tag{6}$$

In the above equation, the first inequality arises from the fact that $\mathbb{E}\|(w_t^k - \overline{w}_{t_0}) - \mathbb{E}(w_t^k - \overline{w}_{t_0})\|_2^2 \leq \mathbb{E}\|(w_t^k - \overline{w}_{t_0})\|_2^2$. The third inequality and last inequality arise from assumption 4 and $\eta_{t_0} \leq 2\eta_{t_0+E} \leq 2\eta_t$. Lemma 1 shows that the variance of the client model parameters is gradually reduced within each local iteration interval I. ■

Lemma 2. *If assumption 4 holds, and η_t is non-increasing with $\eta_t \leq 2\eta_{t+E}$, we have:*

$$\mathbb{E}\|\overline{e}_{t+1} - \overline{a}_{t+1}\|_2^2 \leq \frac{4P}{(P-2B)^2} \cdot \eta_t^2 E^2 G^2$$

Proof. Let us first consider a simple situation. For a group of scalars $S = \{p_0 \leq p_1 \leq ... \leq p_{P-1}\}$, we arbitrarily tamper with the $B < P/2$ numbers and get a new set of scalars $\tilde{S} = \{q_0 \leq q_1 \leq ... \leq q_{P-1}\}$. Then we have:

$$p_{k-B} \leq q_k \leq p_{k+B}, \forall k \in [B, P-B-1] \tag{7}$$

This conclusion is easily obtained by proof by contradiction. If $p_{k-B} > q_k$, it indicts that there are at least $k+1$ numbers in $\tilde{S}$ smaller than p_{k-B}. Therefore, at least $B+1$ numbers in S need to be reduced, which is contrary to the fact that only B numbers can be modified. The proof of $q_k \leq p_{k+B}$ is similar. Besides, we have:

$$\begin{aligned} \mathbb{E}[(\frac{1}{P-2B}\sum_{k=0}^{P-2B-1} p_k - \mu)]^2 &\leq \mathbb{E}[(\frac{1}{P-2B}\sum_{k=0}^{K-1} p_k - \mu)]^2 \\ &\leq \frac{K\sigma^2}{(P-2B)^2} \end{aligned} \tag{8}$$

In the above equation, μ and σ^2 denote the expectation and variance of scalars in S. Similarly, we also have: $\mathbb{E}[(\frac{1}{P-2B}\sum_{k=2B}^{P-1} p_k - \mu)]^2 \leq \frac{K\sigma^2}{(P-2B)^2}$. Combining the above conclusions, for a trimmed mean function of a trimmed rate $\beta = B/P$, we have: $\frac{1}{P-2B}\sum_{k=0}^{P-2B-1} p_k - \mu \leq \mathbf{trmean}_\beta\{q_0, ..., q_{P-1}\} \leq \frac{1}{P-2B}\sum_{k=2B}^{P-1} p_k - \mu$. Then, the variance of trimmed mean can be bounded as:

$$\begin{aligned} &\mathbb{E}(\mathbf{trmean}_\beta\{q_0, ..., q_{P-1}\} - \mu)^2 \\ &\leq \max\{\mathbb{E}(\frac{1}{P-2B}\sum_{k=0}^{P-2B-1} p_k - \mu)^2, \mathbb{E}(\frac{1}{P-2B}\sum_{k=2B}^{P-1} p_k - \mu)^2\} \\ &\leq \frac{P\sigma^2}{(P-2B)^2} \end{aligned} \tag{9}$$

Equation (9) reflects that the estimation error of trimmed mean can be bounded by the variance of the sample. Let σ_i^2 denote the variance of i-th dimension, we have:

$$\begin{aligned}\mathbb{E}\|\overline{e}_{t+1}-\overline{a}_{t+1}\|_2^2 &\leq \sum_{i=0}^{d-1}\frac{P\sigma_i^2}{(P-2B)^2}\\ &\leq \frac{4P}{(P-2B)^2}\cdot\eta_t^2E^2G^2\end{aligned} \tag{10}$$

■

Lemma 3. *In the model aggregation stage, if each client adopts a sparse upload strategy, i.e. randomly selects a parameter server to upload its local model, then when $t+1 \in I_E$, we have:*

- *Unbiased sample estimation:*

$$\mathbb{E}(\overline{a}_{t+1}) = \overline{v}_{t+1}.$$

- *Bounded variance:*

$$\mathbb{E}\|\overline{a}_{t+1}-\overline{v}_{t+1}\|_2^2 \leq \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot\eta_t^2E^2G^2.$$

Proof. For each parameter server i, we know $\mathbb{E}(N_i) = K/P$. Therefore N_i can be regarded as a set that randomly select K/P samples form K clients, which is same as FedAvg with partial device participation. Therefore, according to Lemma 3 and 4 in [17], we know $\mathbb{E}(a_{t+1}^i) = \overline{v}_{t+1}$ and $\mathbb{E}\|\overline{a}_{t+1}^i - \overline{v}_{t+1}\|_2^2 \leq \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot\eta_t^2E^2G^2$. Thus, we have:

$$\begin{aligned}\mathbb{E}\|\overline{a}_{t+1}-\overline{v}_{t+1}\|_2^2 &= \mathbb{E}\|\frac{1}{P}\sum_{i\in\mathcal{P}}\overline{a}_{t+1}^i-\overline{v}_{t+1}\|_2^2\\ &\leq \frac{1}{P}\sum_{i\in\mathcal{P}}\mathbb{E}\|\overline{a}_{t+1}^i-\overline{v}_{t+1}\|_2^2\\ &= \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot\eta_t^2E^2G^2\end{aligned} \tag{11}$$

The transformation in the above formula comes from the convexity of the L_2 norm and the Jensen's inequality.

■

Corollary 4. *If assumptions 4 holds, when $t+1 \in I_E$, $\overline{w}_{t+1}$ can be regarded as a biased estimate of $\overline{v}_{t+1}$. Besides, the estimation error $\|\overline{w}_{t+1}-\overline{v}_{t+1}\|_2^2$ is bounded by $\frac{4P}{(P-2B)^2}\cdot\eta_t^2E^2G^2 + \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot\eta_t^2E^2G^2$.*

C. Analysis for single step mini-batch SGD

Lemma 5. *Assume that assumption 1, 2, 3 and 4 hold. If learning rate η_t is non-increasing with $\eta_t \leq 2\eta_{t+E}$ and $\eta_t \leq \frac{1}{4L}$, with $\Gamma = F^* - \frac{1}{K}\sum_{k=0}^{K-1}F_k^*$, we have:*

$$\begin{aligned}\mathbb{E}\|\overline{v}_{t+1}-w^*\|_2^2 \leq (1-\mu\eta_t)\mathbb{E}\|\overline{w}_t-w^*\|_2^2 + 6L\eta_t^2\Gamma\\ + 8\eta_t^2E^2G^2 + \frac{\eta_t^2}{K}\sum_{k=0}^{K-1}\sigma_k^2.\end{aligned}$$

Proof. Obviously, from equation (5), we know that $\overline{v}_{t+1} = \overline{w}_t - \eta_t g_t$ holds for any time step t. Then, we have:

$$\begin{aligned}\|\overline{v}_{t+1}-w^*\|_2^2 &= \|\overline{w}_t-\eta_tg_t-w^*+\eta_t\overline{g}_t-\eta_t\overline{g}_t\|_2^2\\ &\leq \|\overline{w}_t-\eta_t\overline{g}_t-w^*\|_2^2+\eta_t^2\|\overline{g}_t-g_t\|_2^2\end{aligned} \tag{12}$$

In equation (12), $\|\overline{w}_t-\eta_t\overline{g}_t-w^*\|_2^2$ represents the expected result after a single-step mini-batch SGD, and $\|\overline{g}_t-g_t\|_2^2$ represents the variance introduced by stochastic gradient descend. We first prove that this variance can be bounded. Specifically, if assumption 3 holds, we have:

$$\begin{aligned}\mathbb{E}\|\overline{g}_t-g_t\|_2^2 &= \mathbb{E}\|\frac{1}{K}\sum_{k=0}^{K-1}(\nabla F_k(w_t^k,\xi_t^k)-\nabla F_k(w_t^k))\|_2^2\\ &\leq \frac{1}{K}\sum_{k=0}^{K-1}\|(\nabla F_k(w_t^k,\xi_t^k)-\nabla F_k(w_t^k))\|_2^2\\ &\leq \frac{1}{K}\sum_{k=0}^{K-1}\sigma_k^2\end{aligned} \tag{13}$$

We next analyze the impact of single-step mini-batch SGD.

$$\begin{aligned}&\|\overline{w}_t-\eta_t\overline{g}_t-w^*\|_2^2\\ =&\|\overline{w}_t-w^*\|_2^2+\eta_t^2\|\overline{g}_t\|_2^2-2\eta_t<\overline{w}_t-\overline{g}_t,w^*>\\ \leq&\|\overline{w}_t-w^*\|_2^2+\frac{2L\eta_t^2}{K}\sum_{k=0}^{K-1}(F_k(w_k^t)-F_k^*)-\frac{2\eta_t}{K}\sum_{k=0}^{K-1}\\ &(<\overline{w}_t-w_t^k,\nabla F_k(w_t^k)>+<w_t^k-w^*,\nabla F_k(w_t^k)>)\end{aligned} \tag{14}$$

The above formula uses the fact that if function $F_k(\cdot)$ is L-smooth, we have: $\|\nabla F_k(w)\|_2^2 \leq 2L(F_k(w)-F_k^*)$. By Cauchy-Schwarz inequality and AM-GM inequality, we have:

$$-2<\overline{w}_t-w_t^k,\nabla F_k(w_t^k)> \leq \frac{1}{\eta_t}\|\overline{w}_t-w_t^k\|_2^2+\eta_t\|\nabla F_k(w_t^k)\|_2^2 \tag{15}$$

If assumption 2 holds, we have:

$$\begin{aligned}-2<w_t^k-w^*,\nabla F_k(w_t^k)> \leq -2(F_k(w_t^k)-F_k(w^*))\\ -\mu\|w_t^k-w^*\|_2^2\end{aligned} \tag{16}$$

By combining equation (14), (15) and (16), we have:

$$\begin{aligned}&\|\overline{w}_t-\eta_t\overline{g}_t-w^*\|_2^2\\ \leq&(1-\mu\eta_t)\|\overline{w}_t-w^*\|_2^2+\frac{1}{K}\sum_{k=0}^{K-1}\|\overline{w}_t-w_t^k\|_2^2+\frac{4L\eta_t^2}{K}\\ &\sum_{k=0}^{K-1}(F_k(w_t^k)-F_k^*)-\frac{2\eta_t}{K}\sum_{k=0}^{K-1}(F_k(w_t^k)-F_k(w^*))\end{aligned} \tag{17}$$

In the equation (17), we can see that $\|\overline{w}_t-\eta_t\overline{g}_t-w^*\|_2^2$ is bounded by four terms. The first item represents the recursive relationship, the second item represents the variance of the model parameters, and the third and fourth items reflect the distance from the local optimal solution and the global optimal in t-th round. We next try to bound these items. Firstly, the variance of model parameters can be bounded by $4\eta_tE^2G^2$ according to Lemma 1. Then, let us define $C = \frac{4L\eta_t^2}{K}\sum_{k=0}^{K-1}(F_k(w_t^k)-F_k^*) - \frac{2\eta_t}{K}\sum_{k=0}^{K-1}(F_k(w_t^k)-F_k(w^*))$, we can rewrite C as:

$$C = -\gamma_t\underbrace{\frac{1}{K}\sum_{k=0}^{K-1}(F_k(w_t^k)-F^*)}_{D}+4L\eta_t^2\Gamma \tag{18}$$

, where $\gamma_t = 2\eta_t(1-2L\eta_t)$ and $\Gamma = F^* - \frac{1}{K}\sum_{k=0}^{K-1} F_k^*$ reflects the heterogeneity of data distribution. If the local data are i.i.d, obviously $\mathbb{E}(\Gamma) = 0$. To bound D, we have:

$$\begin{aligned} D &= \frac{1}{K}\sum_{k=0}^{K-1}(F_k(w_t^k) - F_k(\overline{w}_t) + F_k(\overline{w}_t) - F^*) \\ &\geq \frac{1}{K}\sum_{k=0}^{K-1} < \nabla F_k(\overline{w}_t), w_t^k - \overline{w}_t > + (F(\overline{w}_t) - F^*) \\ &\geq \frac{1}{K}\sum_{k=0}^{K-1} < \nabla F_k(\overline{w}_t), w_t^k - \overline{w}_t > \end{aligned} \tag{19}$$

In the above equation, the first inequality comes from Assumption 2, and the second inequality comes from the fact that $F(\overline{w}_t) - F^* \geq 0$. Then, by using AM-GM inequality, and properties of Assumption 1, we have:

$$C \leq 6L\eta_t^2\Gamma + \frac{1}{K}\sum_{k=0}^{K-1}\|\overline{w}_t - w_k^t\|_2^2 \tag{20}$$

Combine Lemma 1, equation (17) and (20), we have:

$$\begin{aligned} \mathbb{E}\|\overline{v}_{t+1} - w^*\|_2^2 \leq (1-\mu\eta_t)\mathbb{E}\|\overline{w}_t - w^*\|_2^2 + 6L\eta_t^2\Gamma \\ + 8\eta_t^2E^2G^2 + \frac{\eta_t^2}{K}\sum_{k=0}^{K-1}\sigma_k^2 \end{aligned} \tag{21}$$

■

D. Convergence Analysis of Fed-MS

Theorem 1. *Assume that assumption 1, 2, 3 and 4 hold, by choosing* $\gamma = \max(\frac{8L}{\mu}, E)$ *and* $\eta_t = \frac{2}{\mu(\gamma+t)}$*, then we have:*

$$\mathbb{E}(F(\overline{w}_t) - F^*) \leq \frac{L}{2\mu(\gamma+t)}(4\triangle + \gamma\mu^2\|\overline{w}_1 - w^*\|^2)$$

, where $\triangle = 6L\Gamma + 8E^2G^2 + \frac{1}{K}\sum_{k=0}^{K-1}\sigma_k^2 + \frac{4P}{(P-2B)^2}\cdot E^2G^2 + \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot E^2G^2$ *and* $\Gamma = F^* - \frac{1}{K}\sum_{k=0}^{K-1}F_k^*$.

Proof. Obviously, we have:

$$\begin{aligned} \|\overline{w}_{t+1} - w^*\|_2^2 &= \|\overline{w}_{t+1} - \overline{v}_{t+1} + \overline{v}_{t+1} - w^*\|_2^2 \\ &\leq \underbrace{\|\overline{w}_{t+1} - \overline{v}_{t+1}\|_2^2}_{E_1} + \underbrace{\|\overline{v}_{t+1} - w^*\|_2^2}_{E_2} \end{aligned} \tag{22}$$

If $t+1 \notin I_E$, $\overline{v}_{t+1}$ is an unbiased estimation of $\overline{w}_{t+1}$. If $t+1 \in I_E$, from corollary 4, we know E_1 can be bounded. By using lemma 5, we have:

$$\begin{aligned} &\mathbb{E}\|\overline{w}_{t+1} - w^*\|_2^2 \\ \leq &(1-\eta_t\mu)\mathbb{E}\|\overline{w}_t - w^*\|_2^2 + 6L\eta_t^2\Gamma + 8\eta_t^2E^2G^2 + \frac{\eta_t^2}{K}\sum_{k=0}^{K-1}\sigma_k^2 \\ &+ \frac{4P}{(P-2B)^2}\cdot\eta_t^2E^2G^2 + \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot\eta_t^2E^2G^2 \\ \triangleq &(1-\eta_t\mu)\mathbb{E}\|\overline{w}_t - w^*\|_2^2 + \eta_t^2\triangle \end{aligned} \tag{23}$$

Equation (23) reflects the recursive relationship of model parameters during the Fed-MS training process. Next, we use this for reduction.

Considering $\eta_t = \frac{\phi}{t+\gamma}$ and $\eta_1 \leq \min(\frac{1}{\mu}, \frac{1}{4L}) = \frac{1}{4L}$ where $\phi > \frac{1}{\mu}$ and $\gamma > 0$, we next to prove that $\mathbb{E}\|\overline{w}_t - w^*\|_2^2 \leq \frac{v}{\gamma+t}$ where $v = \max(\frac{\phi^2\triangle}{\phi\mu-1}, \gamma\cdot\|\overline{w}_0 - w^*\|_2^2)$. When $t=0$, obviously this conclusion holds. Suppose that it holds for a specific time-step $t \geq 1$, we have:

$$\begin{aligned} &\mathbb{E}\|\overline{w}_{t+1} - w^*\|_2^2 \\ &\leq (1-\frac{\phi\mu}{t+\gamma})\frac{v}{t+\gamma} + \frac{\phi^2\triangle}{(t+\gamma)^2} \\ &\leq \frac{(t+\gamma-1)v + \phi^2\triangle}{(t+\gamma)^2} \\ &\leq \frac{v}{t+\gamma+1}\cdot\frac{(t+\gamma-1)v + \phi^2\triangle}{v(t+\gamma)^2} \\ &\leq \frac{v}{t+\gamma+1} \end{aligned} \tag{24}$$

Besides, if $\phi = \frac{2}{\mu}$, $\gamma = \max(8\frac{L}{\mu}, E)$, we have:

$$\begin{aligned} v &\leq \max(\frac{\phi^2\triangle}{\phi\mu-1}, \gamma\cdot\|\overline{w}_0 - w^*\|_2^2) \\ &\leq \frac{\phi^2\triangle}{\phi\mu-1} + \gamma\cdot\|\overline{w}_0 - w^*\|_2^2 \\ &\leq \frac{4\triangle}{\mu^2} + \gamma\cdot\|\overline{w}_0 - w^*\|_2^2 \end{aligned} \tag{25}$$

By the L-smooth assumption, we further have:

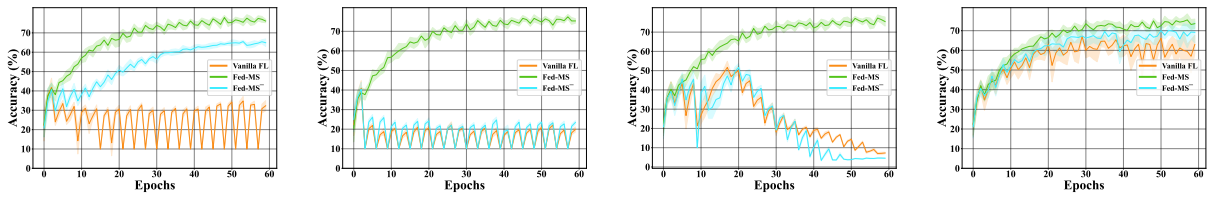
$$\begin{aligned} \mathbb{E}(F(\overline{w}_t) - F^*) &\leq \frac{L}{2}\frac{v}{\gamma+t} \\ &\leq \frac{L}{2\mu(\gamma+t)}(4\triangle + \gamma\cdot\mu^2\|\overline{w}_0 - w^*\|_2^2) \end{aligned} \tag{26}$$

, where $\triangle = 6L\Gamma + 8E^2G^2 + \frac{1}{K}\sum_{k=0}^{K-1}\sigma_k^2 + \frac{4P}{(P-2B)^2}\cdot E^2G^2 + \frac{K-P}{K-1}\cdot\frac{4}{P}\cdot E^2G^2$ and $\Gamma = F^* - \frac{1}{K}\sum_{k=0}^{K-1}F_k^*$. ■

Theorem 1 illustrates the convergence behavior of Fed-MS. Specifically, $\|\overline{w}_0 - w^*\|_2^2$ represents the gap between the global model at the initial time step and the optimal global model, and $\triangle$ represents the error of the algorithm from five aspects. The first term reflects the gap between the global optimal solution and the average local optimal solution, and reflects the non-iid degree of the data to a certain extent. The second and third terms respectively reflect the global gradient bound and the variance of the global model parameters. The fourth term comes from the Byzantine PS, which leads to errors in the estimation of the global model for each client by using trimmed mean. The fifth term comes from the error caused by partial participation, that is, since each parameter server only gets a partial aggregation result, which is different from the global result full clients participation FL. Simultaneously, we can observe that Fed-MS converges at a rate of $\mathcal{O}(1/T)$ in expectation, where T is the number of training rounds.

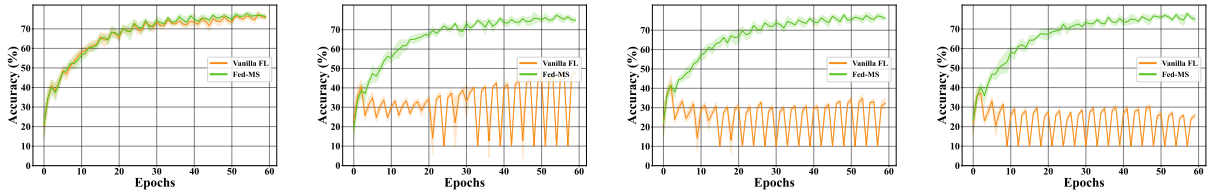
VI. Simulation

In this section, we simulate a FEEL framework with multiple PSs on the edge side and a group of clients on the end side. On each of the client, a visual image classification model is trained. Four Byzantine behaviors are deployed on edge-based PSs, including Noise, Random, Safeguard and Backward attacks [21]. The numerical results reported in this simulation show that our Fed-MS can effectively resist these edge-side Byzantine attacks. At the same time, the impact of the number



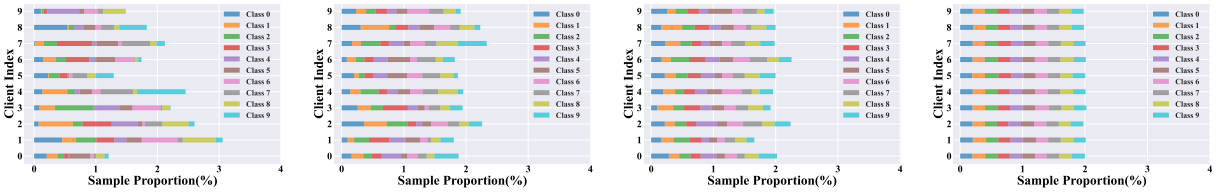
(a) Noise attack (b) Random attack (c) Safeguard attack (d) Backward attack

Fig. 2: Test accuracy versus training epochs under various attack methods



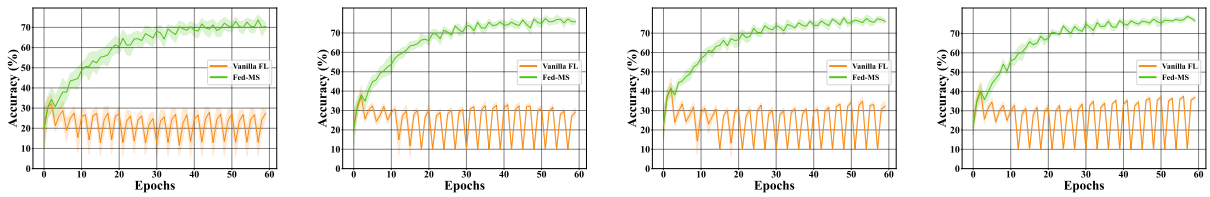
(a) $\epsilon = 0\%$ (b) $\epsilon = 10\%$ (c) $\epsilon = 20\%$ (d) $\epsilon = 30\%$

Fig. 3: Test accuracy versus training epochs when the proportion of Byzantine PSs varies



(a) $D_\alpha = 1$ (b) $D_\alpha = 5$ (c) $D_\alpha = 10$ (d) $D_\alpha = 1000$

Fig. 4: Data distribution of first 10 clients with various D_α



(a) $D_\alpha = 1$ (b) $D_\alpha = 5$ (c) $D_\alpha = 10$ (d) $D_\alpha = 1000$

Fig. 5: Test accuracy versus training epochs with various D_α

of Byzantine parameter servers and data distribution on the model accuracy are explored.

A. Simulation Settings.

In the experiment, we select a widely used image classification dataset CIFAR-10 [36] and adopt MobileNet V2 model [20], a lightweight model focused on mobile and embedded devices, as the training model. All experiments are conducted on a Linux machine equipped with three NVIDIA GeForce RTX 4090s and 192 GB of main memory.

Byzantine Attacks Settings. We implement four common Byzantine attack methods on the server side in our simulation, namely Noise, Random, Safeguard, and Backward [37]. Specifically, the Noise attack introduces a Gaussian noise to the true aggregation result, causing perturbation. The Random attack replaces the genuine aggregation result with a random variable randomly sampled from the interval $[-10, 10]$. The Safeguard attack is an attack method based on reverse gradients, for a Byzantine client i, $\tilde{a}^i_{t+1} = a^i_{t+1} - \gamma g^i_{t+1}$, where $g^i_{t+1} = a^i_{t+1} - a^i_t$ is a pseudo global gradient and the scaling factor γ is set to 0.6. The Backward attack, characterized as a lagging attack, modifies the real aggregation result to that obtained T rounds ago for a Byzantine client i. In mathematical terms, $\tilde{a}^i_{t+1} = a^i_{t+1-T}$. In our simulation, T is set to be 2.

Dataset	CIFAR-10
Model	MobileNet V2
Attack Methods	Noise, Random, Safeguard, Backward
FL Settings	$K = 50, B = 10, E = 3$ $D_\alpha = 1, 5, 10, 1000, \epsilon = 0\%, 10\%, 20\%, 30\%$

TABLE II: A summary of important settings in simulations.

Federated Learning Settings. We consider 50 end clients and 10 edge servers for aggregation. Each end client employs

the Dirichlet function with parameter $D_\alpha \in \{1, 5, 10, 1000\}$ to obtain distinctive subset of CIFAR-10 as its local training dataset, where parameter D_α reflects the heterogeneity of data distribution among clients [38]. A smaller value of D_α implies a greater degree of non-iid data among clients. We also explore different proportions of Byzantine PSs, denoted as ϵ, ranging from 0% to 30%. During the training process, each edge client performs 3 local iterations and 20 global training simultaneously. We record the average test accuracy of the 50 local models on the CIFAR-10 test dataset across varying training epochs. This comprehensive simulation design enables a thorough examination of the impact of Byzantine parameter server proportions and data heterogeneity on the performance of our proposed Fed-MS algorithm.

Some of the important settings mentioned above are summarized in Table II.

B. Numerical results of Fed-MS

In this part, we present a comprehensive numerical result of our Fed-MS. Specifically, we set the proportion of Byzantine PSs $\epsilon = 20\%$ and deploy four distinct attack methods, Noise, Random, Safeguard and Backward on the corresponding Byzantine PSs. Each client employs the Dirichlet function with $D_\alpha = 10$ to obtain a subset of CIFAR-10 for local training and utilize Fed-MS with a trimmed rate $\beta = 0.2$ to defend against adversaries. Additionally, we introduce a variant, denoted as Fed-MS$^-$, incorporating a trimmed rate $\beta = 0.1$ to further explore the impact of trimmed rate. Additionally, the Vanilla FL without Byzantine defense [33] is considered as comparison. The overall performance is graphically depicted in Fig. 2.

The curves in Fig. 2 reveals that Fed-MS effectively withstands these four Byzantine attacks. As the number of training epoch increases, the average test accuracy of the local model exhibits a gradual ascent, peaking at $73\% \sim 76\%$ after 60 training epochs. In stark contrast, Fed-MS$^-$ and Vanilla FL can reach only $8\% \sim 20\%$ after same duration in Random and Safeguard attack. Additionally, as shown in Fig. 2(d), the staled model parameters produced by the Backward attack substantially hinder the convergence speed and final accuracy. But our Fed-MS can effectively remove these outdated parameters, thereby improving the performance. It is worth mentioning that Fed-MS$^-$ demonstrates improvements in test accuracy of approximately $10\% \sim 30\%$ under Noise and Backward attacks compared to Vanilla FL. However, both algorithms exhibit poor performance under Random and Safeguard attacks, yielding a test accuracy of less than 20%. Interestingly, under Backward attack, Fed-MS$^-$ lags behind Vanilla FL by approximately 2%. This simulation demonstrates the efficacy of Fed-MS in resisting Byzantine attacks from the server side and substantiates the importance of setting the trimmed rate β higher than the proportion of Byzantine PSs ϵ for optimal effectiveness.

C. The impact of the proportion of Byzantine PSs

In this part, we explore the impact of varying proportion of Byzantine PSs, denoted by $\epsilon \in \{0\%, 10\%, 20\% \text{ and } 30\%\}$, on the model test accuracy. Specifically, we maintain the attack strategy as Noise, and record the fluctuations in the average test accuracy of the local model for end client on the CIFAR-10 test dataset over training epochs. The simulation results are presented in Fig. 3.

According to the curves in the Fig. 3(a), our Fed-MS and Vanilla FL both converge as the number of training epochs increases, eventually reaching a prediction accuracy of approximately 75%. In other words, in the absence of Byzantine PSs, our Fed-MS has the similar performance with that of Vanilla FL. Comparatively, in Fig. 3(b), 3(c) and 3(d), we observe that, Fed-MS exhibit the identical converge speed and final test accuracy with the Vanilla FL without Byzantine PSs. However, the final test accuracy of Vanilla FL decreases progressively as the proportion of Byzantine PSs increases, dropping from 48% to 25%, which is 27% and 50% lower than that in Fed-MS. Those results demonstrates the resilience of Fed-MS, emphasizing its ability to maintain efficacy even in the presence of varying Byzantine server proportions.

D. The impact of data heterogeneity

In this part, we explore the impact of diverse data distributions on Fed-MS, recognizing that data distribution significantly influences the federated learning performance. We maintain proportion of Byzantine PSs ϵ to 20% and employ Noise for Byzantine attacks, generating varying degrees of data distribution through distinct Dirichlet parameters $D_\alpha \in \{1, 5, 10, 1000\}$. Fig. 4 illustrates the data distribution among the first 10 clients under different D_α. Notably, an increase in D_α leads to a progressively similar distribution of local data among clients. Specifically, when $D_\alpha = 1000$, the data distribution among all clients becomes nearly identical.

The average test accuracy of local model under different D_α is depicted in Fig. 5. The results reveal that the convergence speed and final test accuracy of the local model improve to a certain extent as D_α increases. For example, when $D_\alpha = 1$, the test accuracy after 20 rounds and 60 rounds of training is about 60% and 70%, which is about 9% and 8% lower than those when $D_\alpha = 1000$. These results clearly demonstrate that a more identical data distribution, reflected in higher D_α values, contributes to the enhanced performance of Fed-MS. This rule also applies to Vanilla FL. As D_α increases, the final performance of Vanilla FL moderately improves by approximately 10%, but the accuracy still stays below 40%.

E. Summary of the simulation

In simulation, we evaluate the overall performance of Fed-MS to defend against Byzantine attacks on edge-based PS. Employing four distinct attack methods, our results demonstrate that Fed-MS effectively resists these adversarial behaviors. Additionally, analyses of the proportion of Byzantine PSs and data distribution highlight Fed-MS resilience and effectiveness in diverse scenarios.

VII. Conclusion

In this paper, we address the fault-tolerant problem in federated edge learning with Byzantine parameter servers,

diverging from the prevalent Byzantine-resilient federated learning approaches that rely on reliable PSs to counteract attacks from clients. Our contribution is pioneering as it is the first to tackle security concerns involving unreliable and potentially Byzantine PSs. To counter malicious attacks from Byzantine PSs, we introduce a novel federated edge learning algorithm, termed Fed-MS, leveraging multi-server technique. This algorithm, equipped with a specially designed trimmed-mean-based model filter, enables each client to deduce a feasible global model for its subsequent round of training, even when confronted with Byzantine PS attacks. Moreover, we propose a sparse uploading strategy to enhance the communication efficiency of model aggregation to multiple PSs. When Byzantine PSs constitute the minority, we prove that our Fed-MS achieves convergence speed comparable to state-of-the-art works under non-Byzantine settings. We hope that our work can shed some light on the security problem of FEEL on the edge side. Considering the FEEL problem with both Byzantine PSs and clients will be our work in the future.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62102232, 62122042, 62302247, Shandong Science Fund for Excellent Young Scholars (No.2023HWYQ-007), and Postdoctoral Fellowship Program of CPSF under Grant GZC20231460.

REFERENCES

[1] Y. Guo, Z. Zhao, K. He, S. Lai, J. Xia, and L. Fan, "Efficient and flexible management for industrial internet of things: A federated learning approach," *Computer Networks*, vol. 192, p. 108122, 2021.

[2] Q. Xia, W. Ye, Z. Tao, J. Wu, and Q. Li, "A survey of federated learning for edge computing: Research problems and solutions," *High-Confidence Computing*, vol. 1, no. 1, p. 100008, 2021.

[3] A. Tak and S. Cherkaoui, "Federated edge learning: Design issues and challenges," *IEEE Network*, vol. 35, no. 2, pp. 252–258, 2020.

[4] J. Shi, W. Wan, S. Hu, J. Lu, and L. Y. Zhang, "Challenges and approaches for mitigating byzantine attacks in federated learning," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 139–146.

[5] Z. Chen, P. Tian, W. Liao, and W. Yu, "Towards multi-party targeted model poisoning attacks against federated learning systems," *High-Confidence Computing*, vol. 1, no. 1, p. 100002, 2021.

[6] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *ICML 2018*. PMLR, pp. 5650–5659.

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

[8] S. Huang, Y. Zhou, T. Wang, and Y. Shi, "Byzantine-resilient federated machine learning via over-the-air computation," in *ICC Workshops 2021*, pp. 1–6.

[9] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *NIPS 2017*, vol. 30.

[10] R. Guerraoui, S. Rouault *et al.*, "The hidden vulnerability of distributed learning in byzantium," in *ICML 2018*, pp. 3521–3530.

[11] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," *arXiv preprint arXiv:1909.05125*, 2019.

[12] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *RAID 2020*, pp. 301–316.

[13] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *ACSAC 2016*, pp. 508–519.

[14] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *AISTATS 2020*, pp. 2938–2948.

[15] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" *arXiv preprint arXiv:1911.07963*, 2019.

[16] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021.

[17] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[18] A. Mitra, R. Jaafar, G. J. Pappas, and H. Hassani, "Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients," *NeurIPS 2021*, vol. 34, pp. 14 606–14 619.

[19] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR 2018*, pp. 4510–4520.

[21] S. Li, L. Ju, T. Zhang, E. Ngai, and T. Voigt, "Blades: A unified benchmark suite for byzantine attacks and defenses in federated learning," *arXiv preprint arXiv:2206.05359*, 2023.

[22] T. Wang, Z. Zheng, and F. Lin, "Federated learning framew ork based on trimmed mean aggregation rules," *Available at SSRN 4181353*, 2022.

[23] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *USENIX Security Symposium 2020*, pp. 1605–1622.

[24] X. Wang, D. Dimitriadis, S. Koyejo, and S. Tople, "Invariant aggregator for defending federated backdoor attacks," *arXiv preprint arXiv:2210.01834*, 2022.

[25] A. Hatamizadeh, H. Yin, P. Molchanov *et al.*, "Do gradient inversion attacks make federated learning unsafe?" *IEEE Trans. Medical Imaging*, 2023.

[26] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-iid data," in *2020 IJCNN*, pp. 1–9.

[27] Q. Zhou, S. Guo, H. Lu, L. Li, M. Guo, Y. Sun, and K. Wang, "Falcon: Addressing stragglers in heterogeneous parameter server via multiple parallelism," *IEEE Trans. Computers*, vol. 70, no. 1, pp. 139–155, 2020.

[28] S. Liu, G. Yu, X. Chen, and M. Bennis, "Joint user association and resource allocation for wireless hierarchical federated learning with iid and non-iid data," *IEEE Trans. Wirel. Commun*, vol. 21, no. 10, pp. 7852–7866, 2022.

[29] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *NeurIPS 2020*, vol. 33, pp. 19 586–19 597.

[30] J. Ma, G. Long, T. Zhou, J. Jiang, and C. Zhang, "On the convergence of clustered federated learning," *arXiv preprint arXiv:2202.06187*, 2022.

[31] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, "Splitfed: When federated learning meets split learning," in *AAAI 2022*, vol. 36, no. 8, pp. 8485–8493.

[32] Y. Gao, M. Kim, S. Abuadbba, Y. Kim, C. Thapa, K. Kim, S. A. Camtepe, H. Kim, and S. Nepal, "End-to-end evaluation of federated learning and split learning for internet of things," *arXiv preprint arXiv:2003.13376*, 2020.

[33] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS 2017*. PMLR, pp. 1273–1282.

[34] Y. Zou, D. Yu, P. Hu, J. Yu, X. Cheng, and P. Mohapatra, "Jamming-resilient message dissemination in wireless networks," *IEEE Trans. Mob. Comput.*, 2021.

[35] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

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

[37] S. Li, L. Ju, T. Zhang, E. Ngai, and T. Voigt, "Blades: A unified benchmark suite for byzantine attacks and defenses in federated learning," *arXiv preprint arXiv:2206.05359*, 2023.

[38] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.