BYZANTINE ATTACK ROBUST FEDERATED LEARNING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ECE IŞIK POLAT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF INFORMATION SYSTEMS

SEPTEMBER 2021

BYZANTINE ATTACK ROBUST FEDERATED LEARNING

submitted by **ECE IŞIK POLAT** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems Department, Middle East Technical University** by,

Date: 09.09.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ece Işık Polat

Signature :

ABSTRACT

BYZANTINE ATTACK ROBUST FEDERATED LEARNING

Işık Polat, Ece
M.S., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Altan Koçyiğit

September 2021, 90 pages

In federated learning (FL), collaborators train a global model collectively without sharing their local data. The local model parameters of the collaborators obtained from their local training process are collected on a trusted server to form the global model. In order to preserve privacy, the server has no authority over the local training procedure. Therefore, the global model is vulnerable to attacks such as data poisoning and model poisoning. Even though many defense strategies have been proposed against these attacks, they often make strong assumptions that are not compatible with the characteristics of FL. Moreover, these proposals have not been analyzed thoroughly. In this thesis, I propose an assumption-free defense mechanism called Byzantine Attack Robust Federated Learning (BARFED). BARFED does not make assumption about federated learning setting such as malicious collaborator ratio, the data distributions of the collaborators, and gradient update similarity. BARFED examines the distance between the global model and the local models of the collaborators on a layer basis and decides whether the collaborators will participate in the aggregation rule step phase based on the status of being an outlier. In other words, only the collaborators that are not labeled as outliers in any layer of the model architecture can participate in the aggregation step. I have shown that BARFED provides a robust defense against different attacks by performing comprehensive experiments that cover many aspects such as data distribution and whether attackers are organized or not.

Keywords: Federated Learning, Byzantine Attacks, Label Flipping Attacks

ÖZ

BİZANS SALDIRISINA DAYANIKLI FEDERE ÖĞRENME

Işık Polat, Ece
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Doç. Dr. Altan Koçyiğit

Eylül 2021, 90 sayfa

Federe öğrenmede, katılımcılar yerel verilerini paylaşmadan, toplu olarak, küresel bir modeli eğitirler. Katılımcıların yerel eğitim süreçlerinden elde edilen yerel model parametreleri, global modeli oluşturmak üzere güvenilir bir sunucuda toplanır. Gizliliği korumak adına, güvenilir sunucunun eğitim prosedürü üzerinde hiçbir yetkisi yoktur. Bu nedenle, küresel model, veri zehirlenmesi ve model zehirlenmesi gibi saldırılara karşı savunmasızdır. Bu saldırılara karşı birçok savunma stratejisi önerilmiş olsa da bu stratejiler çoğu zaman federe öğrenmenin karakteristiğiyle uyumlu olmayan güçlü varsayımlarda bulunurlar. Bu çalışmalar çoğunlukla kapsamlı deneysel analizler de yapmamışlardır. Bu tezde, BARFED (Bizans Saldırısına Dayanıklı Federe Öğrenim) adı verilen varsayımdan bağımsız bir savunma mekanizması öneriyorum. BARFED, kötü niyetli katılımcı oranı, katılımcıların verilerinin dağılımları ve gradyan güncelleme benzerliği gibi federe öğrenme ortamı hakkında varsayımda bulunmaz. BARFED, küresel model ile katılımcıların yerel modelleri arasındaki mesafeyi katman bazında inceler ve katılımcıların aykırı değer olması durumuna göre ana model toplama kuralı adımına katılıp katılmayacağına karar verir. Başka bir deyişle, yalnızca model mimarisinin herhangi bir katmanında aykırı değer olarak etiketlenmeyen katılımcılar toplama adımına katılabilir. Data dağılımı, saldırganların organize olup olmadığı gibi birçok yönü kapsayan kapsamlı deneyler yaparak BARFED'in farklı saldırılara karşı güçlü bir savunma sağladığını gösteriyorum.

Anahtar Kelimeler: Federe Öğrenim, Bizans Saldırıları, Etiket Çevirme Saldırıları

To my beloved family

ACKNOWLEDGMENTS

First and foremost, I would like to thank my supervisor, Assoc. Prof. Dr. Altan Koçyiğit for his endless support, valuable guidance, and encouragement.

I would also like to thank my thesis committee members Assoc. Prof. Dr. Tuğba Taşkaya Temizel and Assist. Prof. Dr. Roya Choupani for their valuable recommendations.

I acknowledge the support of The Scientific and Technological Research Council of Turkey (TÜBİTAK) BİDEB for 2210-A Graduate Scholarship Program during my MSc study.

I am also thankful to Özgün Ozan Kılıç for helping me with fixing any bugs.

I am grateful to Gülcan Polat and Ali Polat for their warm hearts and support. Also, I would like to thank my fabulous brother Uğur Polat who always makes me laugh with a look or a word.

I want to express my sincere gratitude to my mother, Saliha Güler, for her unconditional love and support throughout my life. Whenever I need her, she is always there for me.

I want to thank my little daughter, Mavi Polat, for being the sunshine in my life. Her heartwarming smiles, hugs, and kisses make me feel very lucky and hopeful.

With my deepest gratitude, I would like to thank my husband, Görkem Polat, for his unwavering support, love, patience, never-ending encouragement, and being my best friend. It's my luck to have someone like him.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BARFED	Byzantine Attack Robust Federated Learning
CFL	Clustered Federated Learning
eSGD	Edge Stochastic Gradient Descent
FATE	Federated AI Technology Enabler
FedAvg	Federated Averaging
FedPaq	Federated Learning method with Periodic Averaging and Quantization
FL	Federated Learning
FedPer	Federated Learning with Personalization Layer
FSVRG	Federated Stochastic Variance Reduced Gradient
GDPR	General Data Protection Regulation
IID	Independent and Identically Distributed
IoT	Internet of Things
IQR	Inter Quartile Range
LoAdaBoost FedAvg	Loss-based Adaptive Boosting Federated Averaging
LOF	Local Outlier Factor
Non-IID	Non Independent Identically Distributed
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
TFF	TensorFlow Federated

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

In federated learning (FL), a group of collaborators, usually working under the supervision of a trusted server, collectively train a global model without disclosing their local data. Generally, an iterative learning approach is employed. In each round, the trusted server sends the global model parameters to the collaborators. Then the collaborators perform local training processes with their local data and send back parameters of their locally trained models to the server. Lastly, the server aggregates the parameters of the local models to form the global model. These steps are repeated until stopping criteria such as convergence, or adequate performance is met [2, 3].

How to assemble locally trained parameters to form a global model is a well-studied topic in the literature, and several aggregation methods and optimization algorithms that mainly incorporate adaptations of Stochastic Gradient Descent (SGD) have been proposed [2, 4, 5, 6, 7, 8, 9]. These methods most commonly assume that all collaborators in the system are trusted. However, it has been shown that, in the presence of malicious collaborators and attacks, the learning performances of such algorithms may significantly deteriorate.

One of the primary motivations of federated learning is privacy. For this reason, the trusted server is designed in such a way that it has no view and control over the local data and the training processes of the collaborators [10]. However, such a structure built for privacy makes FL systems vulnerable to data poisoning, model poisoning, and backdoor attacks. Malicious collaborators can alter or manipulate their local data to corrupt the global model in the data poisoning attacks. Adding noise to the training data or label flipping attacks are examples of data poisoning attacks [10, 11, 12, 13]. In model poisoning attacks, malicious collaborators modify their model updates. Byzantine attacks, in which malicious collaborators send arbitrary updates, are one of the most well-known model poisoning attacks [10, 12, 14, 15, 16]. The purpose of the backdoor attacks is to affect and disrupt the global model on a specific sub-task. To illustrate, some visual artifacts are added to the training set in order to classify "motorcycles" as "cars" on the global model [3, 11, 12, 17, 18].

In the literature, many studies propose defense approaches suggesting aggregation rules to mitigate the performance loss caused by these attacks [14, 17, 19, 20, 21, 22].

However, these studies usually make assumptions that are not very compatible with practical FL settings [3, 16, 23, 24]. Indeed, an experimental setup where data is not independent and identically distributed (IID) or the malicious collaborators are organized can invalidate the assumptions of such studies and suffers from performance problems. Given that the main motivation of FL is privacy, partial or full knowledge defense strategies that are based on examining local datasets and local training processes, e.g., defense methodologies that are using data sanitization against backdoor attacks) are not very realistic. Therefore, analyzing and developing defense methods for realistic FL environments is an important area of research.

1.2 Objectives and Scope

In this thesis, I propose a defense mechanism called Byzantine Attack Robust Federated Learning (BARFED) that does not make any unrealistic assumptions about FL settings. BARFED is applicable for training multi-layer neural networks (NNs) by using FL. BARFED extends FederatedAveraging (FedAvg) [2] algorithm to consider which collaborators are eligible to participate in an aggregation step. BARFED is primarily designed against Byzantine attacks, but the algorithm's performance is also tested against label flipping attacks.

My proposed method (BARFED) adopts the boxplot elimination method and performs a distance-based layer-wise outlier detection approach to detect unreliable collaborators in the FL system. If a collaborator is labeled as an outlier in any layer of the model architecture, even if other layers are in the safe range, the collaborator is discarded from the aggregation rule calculation. The primary motivation here is that if a collaborator is considered malicious, other updates from the same collaborator should not be trusted.

BARFED does not need to know the ratio of malicious collaborators or even if there is an attack on the FL system or not. Yet, it does not make any assumption on whether the collaborators' data are IID or non-IID.

1.3 Contributions and Novelties

The main contributions of this thesis are as follows:

- I propose an assumption-free Byzantine attack-resistant federated averaging method called BARFED. BARFED does not make any unrealistic assumptions that contradict the FL nature, such as a certain data distribution, update similarities of collaborators, or knowledge about the malicious collaborator ratio in the system.

- I conduct extensive experiments to investigate the performance of the BARFED. In these experiments, I investigate factors such as different types of attacks, organized/independent attacks, and IID/non-IID data distributions on three different datasets.

- I show that the other defense algorithms that I evaluated could resist attacks in IID cases, while they perform only a little improvement or worsens the performance degradation in Non-IID settings. Contrary to other methods, my proposed method performs well in both IID and non-IID situations and defends the global model from various attacks under any experimental setting.

1.4 Thesis Outline

The rest of the thesis is organized as below.

The background information about federated learning is presented in Chapter 2. The related work in the literature is reviewed in Chapter 3. The detailed methodology of the proposed approach is presented in Chapter 4. The details of the experimental design are given in Chapter 5. The results of the experiments and effectiveness of the proposed method are discussed in Chapter 6. The conclusion of this thesis and future work are given in Chapter 7.

CHAPTER 2

BACKGROUND

In this chapter, an overview of the federated learning concept is presented. In the first section, the basics of federated learning are introduced, and the difference between the distributed learning and FL is explained. In the second section, the types of federated learning are summarized. Then, prominent aggregation methods and optimization algorithms are given. Lastly, open-source frameworks for federated learning are summarized.

2.1 Federated Learning with Basics

Smart devices such as phones, tablets, and smartwatches have become an integral part of many people's lives. These devices are capable of generating large amounts of useful data thanks to their powerful sensors and rich user interactions. Such data may be turned into value by using machine learning techniques. Conventional techniques that may be employed for this purpose require collecting data from these sources and processing them in a central place. However, the sensitive nature of this data comes with some risks and responsibilities in terms of privacy and data security [2, 5]. Indeed, data security and privacy concerns have been an arising topic recently. Therefore, some regulations and legislation have been proposed by policymakers. For example, the General Data Protection Regulation (GDPR) [25, 26] has been presented in Europe. In addition, a White House report [27] and Consumer Privacy Bill of Rights [28] that focused on the privacy of consumers have been proposed [2, 29].

The concept of FL was first introduced in 2016 to answer these privacy and security concerns. Basically, collaborators train a global model collectively without disclosing their local data, and learning is essentially performed in a decentralized manner [2]. Figure 1 illustrates the main workflow of a sample FL application. Usually, FL employs an iterative approach. In each iteration, the trusted server broadcasts the current version of the model (step 1), the collaborators use their local data to improve the model (step 2) and send the updated model to the server (step 3) which aggregates the updated models to form the next version of the global model (step 4).

Federated learning employs the participants' processing, storage and communication resources in the learning process. Therefore, the learning process is executed in a distributed manner. However, federated learning should not be confused with

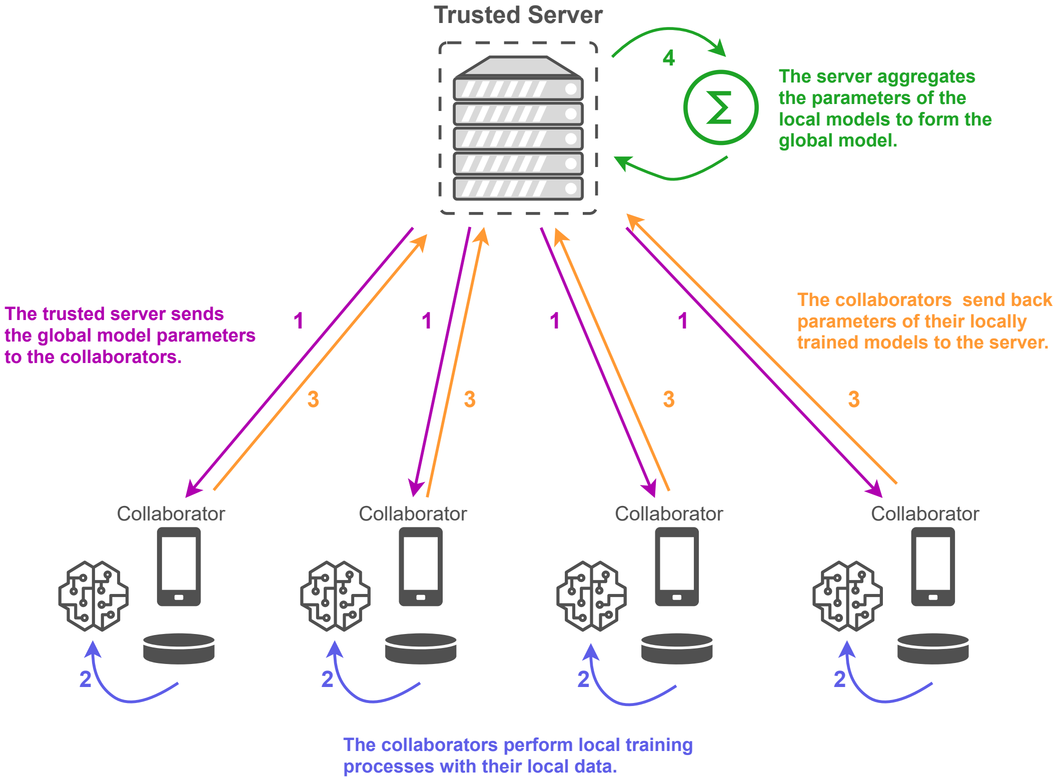
Trusted Server
4
Σ
The server aggregates the parameters of the local models to form the global model.
The trusted server sends the global model parameters to the collaborators.
1
1
1
1
3
3
3
3
The collaborators send back parameters of their locally trained models to the server.
Collaborator
Collaborator
Collaborator
Collaborator
2
2
2
2
The collaborators perform local training processes with their local data.

Figure 1: FL architecture.

distributed learning. One of the main motivations of distributed learning is taking advantage of rich computational resources with the increasing availability of grid computing technology and multiprocessors [30]. In this context, to take advantage of computational resources, the central server can access all training data and divide the data into subsets that have similar data distributions, typically in an IID fashion. Then, these subsets are sent to nodes for parallel training. In other words, the server decides how to divide the training data and which collaborator nodes train these data subsets [29, 31]. On the other hand, the main motivation of FL is privacy, and the trusted server has no access to the training data. Therefore, in FL, the server cannot decide how data division will be applied, and the data distribution across the collaborators is naturally non-independent and identically distributed (non-IID) as the collaborators have different data distributions and data amounts based on differences in their user information, time window, device capabilities, or geographic location [31, 3].

There are two federated settings proposed in [3]: cross-silo and cross-device. The collaborators are from different organizations or institutions such as hospitals, banks, or e-commerce firms in the cross-silo setting. In cross-silo settings, the number of collaborators is much smaller than that of the cross-device settings. Moreover, the identities of the participants are usually known by all entities in the setting, collaborators usually have rich resources, and learning iterations can be done synchronously as all collaborators are almost always available [3]. On the other hand, collaborators are probably mobile devices, smartphones, or IoT devices in the cross-device setting. The identities of the collaborators are not known by other entities taking part in the learning process. Participants have limited resources; hence their collaboration can only be feasible when they are online, they have a free network connection (e.g., WiFi), and are plugged in. Therefore, updates generally cannot be done synchronously since only a portion of all collaborators is available simultaneously [3].

The defense method I propose in Chapter 4 can be applied in both cross-silo and cross-device settings.

2.2 Federated Learning Types

The FL concept is categorized into three types. These are Horizontal FL, Vertical FL, and Federated Transfer Learning [32].

In Horizontal FL, datasets share the same feature space over different sample spaces. For example, the users served by the two job search platforms may be very different due to geographic separation, or the intersection set of these users may be very small. But basically, because the work done by these two platforms is very similar to each other, the feature space of their data sets may be considered the same [32].

In Vertical FL, datasets share the same sample space over different feature spaces. For example, considering an online job search platform and an online education platform serving in the same country, the people in their datasets are likely to cover most of the users in the region, and the intersection set of these users is probably large. While the online job search platform has information such as most wanted qualifications, the most commonly searched keyword, trends in business, the online education platform has information about the area and the people they serve [32].

In Federated Transfer Learning, datasets differ in terms of both samples and feature spaces. For example, consider an online education platform operating in China and an online job search platform company operating in the United States. The intersection of the user groups of the two institutions is tiny due to possible geographic constraints. Likewise, a small part of the attribute area of the data sets overlaps due to different business strategies. In this style, federated transfer-learning techniques can be applied when solutions need to be produced for all sample and attribute areas [32].

The defense method that I propose in Chapter4 is applicable only for horizontal federated learning types.

2.3 Application Areas of Federated Learning

The sectors and domains where data ownership, data security, privacy, and intellectual property rights are a concern are promising areas of application for federated learning [32]. For instance, in the healthcare domain, data is located in different places such as hospitals, private clinics, or smart devices. Due to the scalability and privacy concerns, it is not possible to collect and process all these data in a single database. Therefore, FL has an important application area in the healthcare and medical domain [33, 34]. Recently, FL applications in the healthcare and medical domain attracted researchers, and several studies have been presented [35, 36, 37, 38, 39, 40, 41, 42].

Recommendation systems ([43, 44, 45]), transportation ([46, 47, 48, 49, 50]), finance ([51, 52]), speech recognition ([53, 54, 55, 56]), computer vision ([57, 58]) and natural language processing ([59, 60, 61, 62]) are other areas federated learning approaches are studied.

2.4 Algorithms and Methods

Several methods and optimization algorithms have been proposed for the global model aggregation. Some of them are introduced below.

FederatedAveraging (FedAvg): FedAvg is one of the most widely used methods [63]. Hence, I base my model on FedAvg. This method is mainly used for training a neural network, but it can be adapted to other parameterized models. As demonstrated in the Algorithm 1, in this method, collaborators train their local models in parallel with a given number of epochs and send only the parameters of their local models to the trusted server. Then the weighted average of local model parameters of collaborators is calculated on the trusted server, and this calculation is used as new parameters of the global model for the next FL round [2].

Algorithm 1 FedAvg. T is the number of rounds, P is the number of collaborators, s_p is the relative sample size for collaborator p, J is the number of local epochs, η is the learning rate, ℓ is the cost function, x is the training data, and y is the target.

```
procedure SERVERUPDATE
    initialize weights, $w_1$
    for each round $t = 1, 2, ...T$ do
        send $w_t$ (main model weights) to the collaborators
        for each collaborator $p = 1, 2, ...P$ do in parallel
            $w_{t+1}^p \leftarrow$ COLLABORATORUPDATE($p, w_t$)        ▷ Local Training
        end for

        $w_{t+1} \leftarrow \sum_{p=1}^{P} s_p \times w_{t+1}^p$        ▷ aggregation via averaging

procedure COLLABORATORUPDATE($p, w$)
    for each step $j = 1, 2, ...J$ do
        $w_{t+1} \leftarrow w_t - \eta \nabla \ell(x^p, y^p; w)$
    return $w$ to the server
```

Federated Stochastic Variance Reduced Gradient(FSVRG): The loss function of Stochastic Gradient Descent (SGD) has a slow convergence rate due to variance in its nature. The Stochastic Variance Reduced Gradient (SVRG) method aims to increase the convergence speed by decreasing this variance [64]. Federated Stochastic Variance Reduced Gradient is the version of this method adapted to federated learning applications. FSVRG trains the local models by repeating the data in the devices with permutation and adjusts the learning rate inversely proportional to the data size [5].

FedProx: This method modifies FedAvg by adding a proximal term to the loss function for more stable convergence in cases of statistical heterogeneity. FedAvg can be seen as a special case of FedProx in which the proximal term μ is set to zero. FedProx encourages collaborators to behave better by restricting the local updates of collaborators to be closer to the current global model. It is independent of assumptions such as IID data distribution, and all devices are active (i.e., synchronous case) [65].

Federated Learning with Personalization Layers (FedPer): In this method, ***base*** layers are trained locally in the collaborators by using their own data. The successive layers, namely ***personalization*** layers, are trained collaboratively on the trusted server according to the outputs of the base layers and the target labels for training examples. FedPer provides *base* + *personalization* deep learning architecture to get rid of the effects of statistical heterogeneity and provide the possibility of customizing the federated learning concept [1]. Figure 2 demonstrates the architecture of FedPer method.

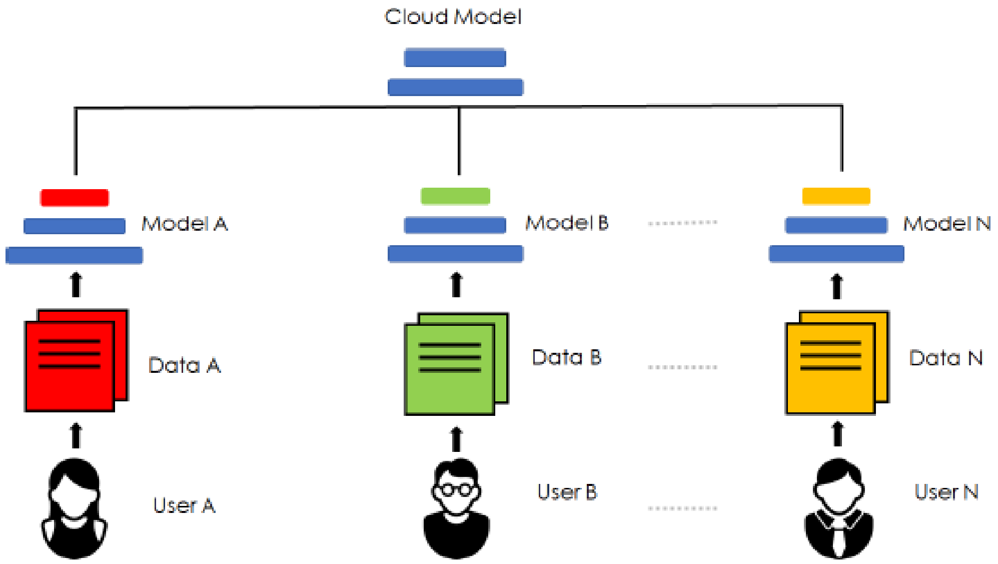

Figure 2: The architecture of FedPer (adapted from [1]).

Federated Learning method with Periodic Averaging and Quantization (FedPaq): This method provides an asynchronous approach such that only some of the collaborators participate in the local model training step. Since the communication bandwidth between the collaborators and trusted server is limited, quantization operators are used for reducing the size of the parameters sent, and these parameters are averaged on the server periodically, i.e., collaborators sent local model parameters after a number of local updates for global model aggregation [66].

Astraea: In this method, except for collaborators and the trusted server, a third identity called a mediator is introduced to the system. Unbalanced data distribution causes degradation in the performance of federated models. Therefore, collaborators send data distributions to the server before the training process starts. Then collaborators are organized for data balancing and rescheduling operations by mediators. Moreover, collaborators perform data augmentation methods such as random rotation and shifting to minority classes in order to increase performance [4, 29].

Loss-based Adaptive Boosting Federated Averaging (LoAdaBoost FedAvg): In this method, the collaborators train their local model in parallel, but before sending their parameters to the trusted server, the current loss is compared with the previous round's median loss. If there is no improvement in loss, these collaborators are seen as weak learners and are made to repeat the local training process [38].

Edge Stochastic Gradient Descent (eSGD): This method aims to significantly reduce communication costs by sending only the important gradients to the trusted server, based on the observation that most of the parameters are sparse and close to zero in an artificial neural network model. The eSGD method determines important gradients by examining parameters that have large weights or improves the loss function compared to the previous FL round and only sends these gradients to the central server for global model aggregation [67, 68].

2.5 Open Source Frameworks for Federated Learning

There are several open-source frameworks and libraries for federated learning. Some of them are listed below.

TensorFlow Federated (TFF): TFF[69] is a TensorFlow [70] based open source framework developed by Google. It enables machine learning and other computations on decentralized data.

PySyft: PySyft is a privacy-preserving Python library compatible with TensorFlow and PyTorch. Techniques related to federated learning, differential privacy, cryptographic computing, and homomorphic encryption can be implemented with PySyft [71, 72].

Clara Training Framework: Clara Training Framework is developed by NVIDIA specifically for the medical imaging field and provides solutions for cross-silo federated learning where collaborators are from different organizations and institutions [73].

OpenFL: OpenFL is a Python library for federated learning applications and developed by Intel Labs. OpenFL is framework agnostic, i.e., it is compatible with both TensorFlow and PyTorch [74].

LEAF: LEAF is an open-source benchmark for federated settings. LEAF provides facilities such as data sets, statistical and systemic metric settings, and reference applications that can be used for federated learning [75].

Flower: Flower is a federated learning framework with customizable configurations, and it is designed for advance FL research. Flower also aims to reduce the gap between production and research [76].

Federated AI Technology Enabler (FATE): Federated AI Technology Enabler is an open-source project for secure computing that is initiated by the Webank. FATE provides a modular, scalable modeling line, clear visual interface, flexible timing system, and operational performance [77].

PaddleFL: PaddleFL is an open-source framework that enables the application of federated learning in many areas such as recommendation systems, natıral language processing, and computer vision [78, 79].

In this thesis, as I need some flexibility to implement certain types of attacks in various scenarios, experiments have been carried out on my own code base built on PyTorch [80] platform.

CHAPTER 3

RELATED WORK

The defense strategies against attacks in FL is a heavily studied research topic recently.

In this chapter, the potential attacks and defense against such attacks suggested in the literature are reviewed. Defense against attacks in FL is an important and heavily studied research topic. Although these mechanisms have some commonalities, I group them according to the main idea employed, such as outlier elimination, distance-based, and similarity-based approaches.

3.1 Security Problems and Attacks in FL

FL suffers from privacy and security issues. The malicious collaborators can cause serious privacy and security issues in FL. For example, the malicious collaborators may violate privacy by exploiting sensitive information from the global model updates. Moreover, the malicious collaborators can obtain the global model even if they do not participate in the local training process. Also, they can disrupt the performance of the global model thanks to poisoning or backdoor attacks. The attack types in FL can be grouped according to the purpose and working mechanism of the malicious collaborators. The main types of attacks are listed below.

Data Poisoning Attacks: In data poisoning attacks, malicious collaborators alter or modify the training data. In this attack type, malicious collaborators may choose to modify all or only a portion of the training data. Label flipping attacks where the actual label values are replaced with other target values are one of the data poisoning attacks. Another example of data poisoning attacks is adding noise to the data, i.e., before the local training process, the malicious collaborators add noise to their local data, and then they start the training [10, 11, 12, 13].

Model Poisoning Attacks: The model parameters are manipulated by the malicious collaborators in the model poisoning attacks. The malicious collaborators can decide the proportion of parameter updates to be changed. Byzantine attacks and adaptive attacks are examples of model poisoning attacks. In Byzantine attacks, malicious collaborators send arbitrary random model parameters to the server for corrupting global model [10, 12, 14, 15]. Adaptive attacks are advanced variations of Byzantine attacks. Generally, in adaptive attacks, malicious collaborators perform local training steps and learn parameters. Then they use these parameters for well-tuned attacks instead of sending random updates [16, 23].

Backdoor Attacks: Backdoor attacks aim to corrupt the global model on a specific sub-task; that is, the malicious collaborators add some visual artifacts to the training data and deceive global model to classify motorcycles as cars. Since backdoor attacks focused on a specific subtask, they are known as targeted attacks, while data poisoning attacks and model poisoning attacks are untargeted attacks [17, 11, 3, 12, 18].

Inference Attacks: At the beginning of each FL round, the trusted server sends the global model parameters to the collaborators who will participate in the training step. In inference attacks, malicious collaborators aim to exploit the sensitive information of the other collaborators by examining differences and the changes in the global model parameters. In other words, the malicious collaborators may recover the original data set and class labels by performing reverse engineering to the shared parameters and gradients [81, 82].

3.2 Distance Based and Outlier Elimination Based Approaches

In [22], instead of taking the average of the gradients directly in FedAvg, two alternative aggregation methods are proposed to avoid Byzantine attacks. They are namely trimmed mean and coordinate wise median. In the trimmed mean method, β, the ratio of malicious collaborators in each FL round must be known. Thus, the local model parameters from each collaborator are ordered from smallest to largest. Then, the smallest and highest β fraction of values are trimmed and averaged. In other words, beta smallest and beta largest parameters are discarded from the global model aggregation calculation. On the other hand, global model parameters are set as the median of the local models' parameters in the coordinate-wise median. The MNIST data set was used in this study, and data were distributed to the collaborators in an IID fashion. Byzantine attacks were designed to send moderate values to the trusted server instead of adding extreme values to the gradients. Therefore, the actual label y has been replaced by 9-y, i.e., 2 has been replaced with 7. However, this attack scenario designed for the Byzantine attack simply coincides with the label flipping attack. Yet, both the trimmed mean and the coordinate-wise median were able to provide improvement in accuracy compared to vanilla FedAvg.

In [14], a method called KRUM is proposed. KRUM combines the majority-based and square-distance methods. In each round, KRUM chooses one of the local models that is closest to the other models. Then, it sets the parameters of the chosen local model as the global model. KRUM needs to know the malicious collaborator ratio in the system. Let n be the total number of collaborators and m be the number of malicious collaborators; for each local model parameter, KRUM calculates $n - m - 2$ closest models based on the Euclidean distance between each model parameter. Later, based on the sum of squares of these distances, a score is generated for each local model, and the local model with the smallest score is selected as the global model. Even if the local model chosen might be received from a malicious collaborator, it is believed that its impact will be limited since it is close to the trusted collaborators. For the experiments, MNIST, SpamBase datasets were distributed IID to the collaborators, and disrupted gradient updates were generated from Normal distribution with mean=0 and standard deviation=200. Krum could tolerate Byzantine attacks up to 33% malicious collaborator ratio; however, when there is no malicious collaborator in the system, vanilla FedAvg(No defense case) results in better performance.

In [15], a method called Bulyan is proposed. Bulyan is a Byzantine resilient aggregation method and is an adaptation of KRUM. A single model parameter can seriously affect the Euclidean distance between two models. For this reason, some extreme and abnormal model parameters may cause performance degradation for KRUM. Consequently, Bulyan combines KRUM and trimmed mean approaches against Byzantine attacks. Similar to KRUM, this method also requires information about the number of malicious collaborators in the system. In KRUM, the local model that is closest to the others is chosen as the new global model. In Bulyan, for each local model parameter, p_i, KRUM is executed and at most $n - 2m$ local models are chosen. Then, for the p_i parameter of the global model in the new round, the p_i parameters of these local models are averaged. Since Bulyan applies KRUM for many times for each model parameter separately, it is considered as not scalable [16]. The experiments show that Bulyan achieved higher accuracy and lower test error than the KRUM. Bulyan could get performance similar to no attack case in terms of accuracy and error rate.

3.3 Similarity Based Defense Approaches

In [83], two-dimensional visualization is performed by applying Principal Component Analysis (PCA) to the parameters in the last layer of the model. Then, through this visualization, a distinction is made between reliable and attacker collaborators. This defense mechanism is based on the assumption that updates from malicious collaborators have unique characteristics, while updates from trusted collaborators are similar to each other. For this reason, the experiments were carried out with IID data distribution, and hence suspicious behaviors could be detected easily. Even if it is not explicitly stated in this study, the decision of attacker collaborators requires human intervention. There is no automated process after the visualization step. It cannot be said that this is a very realistic case in the FL setting. In the experiments, CIFAR10 and Fashion MNIST datasets were used and the performance of the defense strategy was evaluated against label flipping attacks. It has shown that the proposed attack could identify malicious collaborators.

The method proposed in [12] adopts regular Clustered Federated Learning (CFL) against Byzantine attacks. The method clusters the collaborators based on the pairwise cosine similarities of gradient updates. Before the aggregation step, the trusted server performs clustering based on the computed cosine similarities. Then cross similarity between cluster candidates is calculated. When the cross similarity between the cluster candidates is lower than a certain threshold, i.e., clusters candidates are dissimilar enough, the main cluster is divided into the sub-clusters. In regular CFL, the model aggregation step is executed separately for each cluster. However, in byzantine robust

CFL, the collaborators that are in the largest clusters are labeled as trusted, and the rest of the collaborators are labeled as malicious. Hence, only elements of a benign cluster participate in the aggregation rule. In order to evaluate the method, three different attack types are examined: label flipping attacks, noisy data, and Byzantine attacks. The proposed method provided significant improvement in accuracy scores. The accuracy scores obtained in Byzantine attacks are lower than the accuracy scores obtained in noisy data and label flipping attacks.

In [84], a defense method called FLTrust has been proposed against Byzantine attacks. In Byzantine attacks, the attackers can reverse the direction of the updates and increase the magnitude of the updates to increase the effect of the attack. This method aims to avoid these two possibilities. Therefore, the trusted server collects a root dataset for itself and assumes that there is no data poisoning in this root dataset. While the collaborators train their local models with their own local datasets, the trusted server trains its own model on the root dataset. In the aggregation step, the server checks how the directions of the updates from the collaborators differ from the direction of the model it trains with its own data and assigns a trust score to each collaborator in direct proportion to this similarity. Then, the trusted server normalizes the magnitudes of the updates from collaborators and scales them to have the same magnitude of its own update. Finally, it takes the weighted average of local model parameters according to their trust score. The proposed method could tolerate adaptive attacks even if with a large malicious collaborator ratio and provided an error rate like FedAvg when there is no attack case.

In [11], the proposed method FoolsGold makes an assumption on the gradient update similarity. The fundamental assumption behind the FoolsGold is that the data distribution of trusted collaborators is unique; hence, the gradient updates of trusted collaborators vary. On the other hand, it assumes that malicious collaborators share a common purpose, and because of this reason, their gradient updates are more similar. Accordingly, the method behaves collaborators that have similar updates as malicious and decrease their learning rate to reduce their effect on the global model. MNIST, VGGFace2, Cup99, and Amazon Reviews datasets were used in the experiments to test label flipping and backdoor attacks scenarios. The experiments were performed in the non-IID data distribution so that malicious collaborators could be detected easily.

Although FoolsGold ([11]) offers a defense strategy against Byzantine attacks, FoolsGold works only where only trusted participants have unique updates and malicious participants have similar updates, and FLTrust expects similar updates from trusted collaborators. Therefore, it can be said that the similarity-based methods can only cover a part of the experimental space. For example, FoolsGold cannot perform adequately in the IID setting where trusted participants will also make similar updates. Similarly, for both IID and non-IID cases, FoolsGold is not able to distinguish trusted and malicious collaborators when malicious collaborators send different arbitrary updates. On the other hand, the methods proposed in [83] and byzantine robust CFL [12] assume that trusted collaborators have similar updates performed experiments in the IID setting to identify malicious collaborators quickly.

There are many proposed defense strategies against Byzantine attacks, but they mainly made some unrealistic assumptions. Most of the methodologies assume that the data distribution of collaborators is IID ([14, 15, 12, 83, 22]). However, Non-IID data distribution of collaborators is one of the main characteristics of FL ([3, 2]), the existing defense strategies against Byzantine attacks are not able to perform as well as in the Non-IID cases [17]. Moreover, some of the methods ([22, 14, 15] make another unrealistic assumption about knowing malicious collaborator ratio that is not compatible with FL nature. Considering all these factors, I have proposed a method that does not make assumptions about data distribution, update similarity, and malicious collaborator information.

CHAPTER 4

METHODOLOGY

My proposed method is a variation of Federated Averaging(FedAvg) [2] which is one of the most widely used and adopted algorithms in FL [63, 3]. In FedAvg, the trusted server sends the global model to the collaborators at the beginning of each FL round. Collaborators train and update the model with their local data. During the local training phase, collaborators do not share any data with each other and the server. Only the parameters of locally trained models are sent to the server. Finally, the trusted server aggregates these local models to form a potentially improved version of the global model by averaging the parameters(weights) received from the collaborators. In other words, the collaborators train a global model under the orchestration and control of the trusted server without sharing their local data [2, 3].

If the global model converges and there are no attacks in a standard FL setting, the local models of the collaborators are unlikely to drift apart from the global model. On the contrary, the local models of malicious collaborators are likely to move far away from the trusted ones, and the global model in the presence of attacks [10]. From this point of view, I incorporate my proposed method just before the aggregation step of FedAvg to decide which collaborators should be considered in the aggregation step. The main idea of my method is to discriminate reliable collaborators and outliers by evaluating layerwise distances to the respective layers of the latest global model. In this way, the collaborators marked as reliable could participate in the aggregation step, while the other collaborators marked as outliers are discarded from the calculation.

I use the boxplot outlier elimination method [85] to detect outliers. In the boxplot outlier elimination method, for each layer, the distances between the global model and participants' updates are sorted in ascending order, and Q_1 (1^{st} quartile) and Q_3 (3^{rd} quartile) are determined. Q_1 is equivalent to the 25^{th} percentile, i.e, represents the point at which 25% of the data falls below this value. In a similar way, the Q_3 is equivalent to the 75^{th} percentile. Then, Inter Quartile Range (IQR) is calculated by subtracting Q_1 from Q_3 (i.e., $IQR = Q_3 - Q_1$). Later, as shown in Figure 3, the minimum threshold is determined by subtracting $1.5 \times IQR$ from Q_1 (i.e., $Q_1 - 1.5 \times IQR$) and the maximum threshold is determined by adding $1.5 \times IQR$ to the Q_3 (i.e., $Q_3 + 1.5 \times IQR$).

In order to illustrate the rationale and the motivation behind the use of an outlier detection mechanism in defense against attacks, I carried out a simple experiment. In this experiment, a label flipping attack is performed on MNIST-2NN architecture when there are 100 collaborators with malicious collaborator ratio is 20% and the

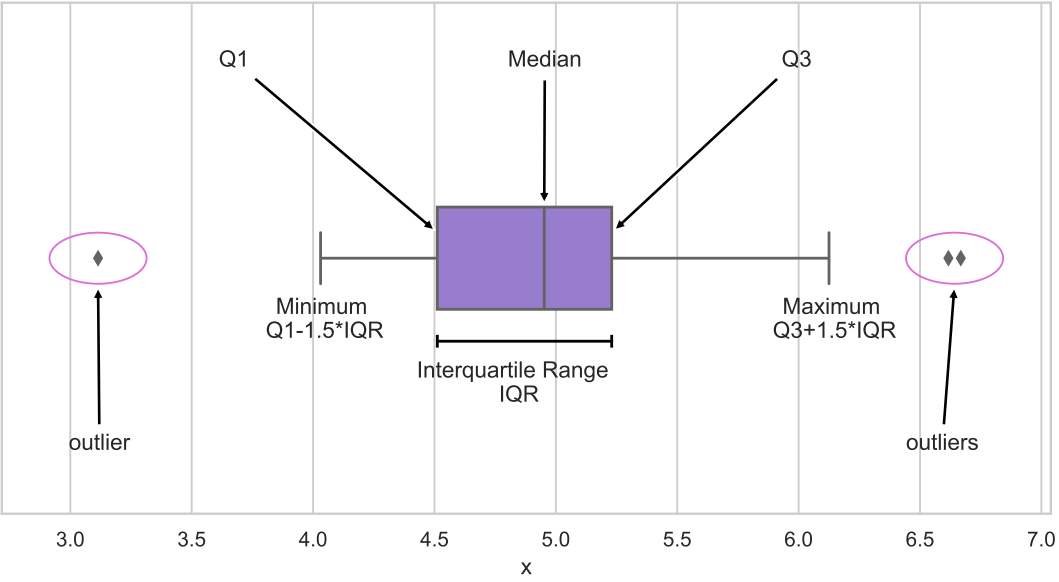

Figure 3: Box plot elimination.

data distribution of collaborators is IID. Figure 4 shows the layerwise distances of a randomly selected normal collaborator and a randomly selected malicious collaborator to the global model for each communication round. As it can be seen from the figure, in all rounds, the distances of the trusted collaborator to the global model for each layer remain inside the aggregation interval (highlighted for each layer, $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$). In contrast, the distances of the malicious collaborator are in the outlier region. BARFED is mainly based on making this discrimination based on distances. The circles demonstrate the distance between the randomly selected collaborator and the global model for each layer of the model architecture. The shaded areas are determined according to the updated models returned by the collaborators, and they indicate the upper and the lower bounds, i.e., safety region, that are used to determine outliers. Moreover, the further experiments (see Figure 28 and Figure 27) have shown that the rate of collaborators labeled as unreliable by BARFED is very close to the rate of actual malicious collaborators.

Algorithm 2 gives the details of my proposed defense method BARFED. This algorithm assumes that the global model consists of multiple NN layers. Figure 5 represents an exemplary model architecture. Here, each colored rectangle represents the weights of the respective layer. Weights of each layer are converted into 1D vectors, and layerwise distances to the global model are calculated in the proposed method. The algorithm determines the outlier status of a collaborator by considering outlier statuses of each of the layers returned by the collaborator. If a layer is determined as an outlier, the collaborator is determined as malicious, and its model is discarded from model aggregation.

In **procedure** COLLABORATORUPDATE, the collaborators receive global model parameters and set them as their local model parameters. After that, the collaborators

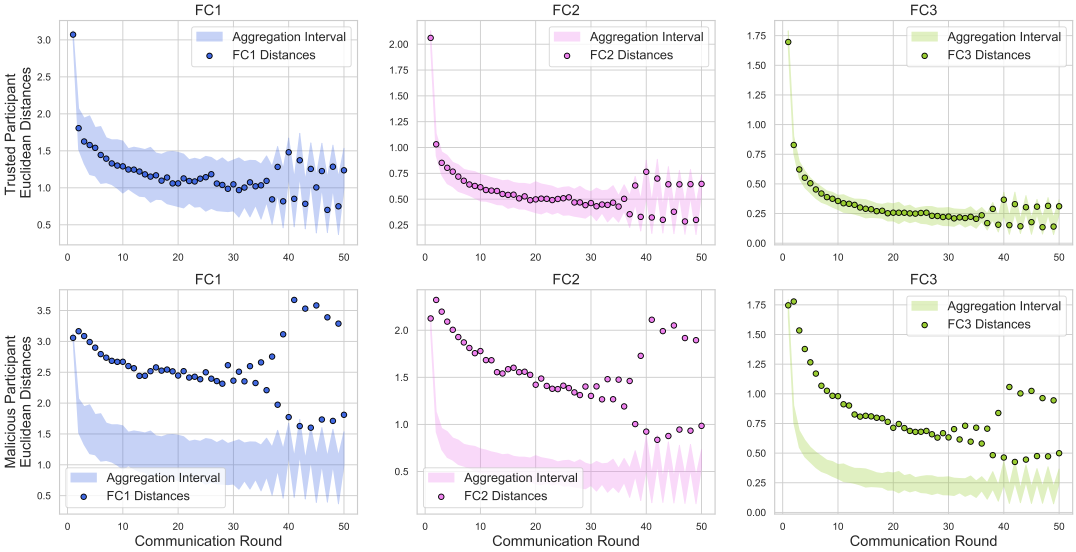

Figure 4: Distances of a randomly selected normal collaborator (upper row) and a randomly selected malicious collaborator to the global model under label flipping attack on MNIST dataset.

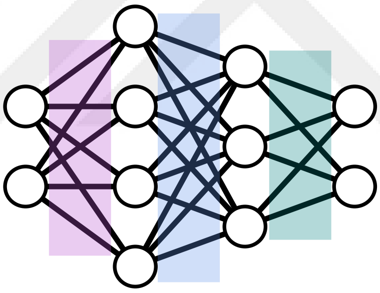

Figure 5: An exemplary model architecture.

update their models by performing a local training process with their local data and then send their updated model parameters to the server for the aggregation step.

In **procedure** SERVERUPDATE, the server receives updates $w_{t+1}^{p_1}, w_{t+1}^{p_2}, \ldots, w_{t+1}^{p_n}$ from collaborators at the server at any round t. The L^2 norm distances [1] between parameters of each collaborator and global model parameters are calculated layer by layer. In other words, the Euclidean distance of each collaborator's i^{th} layer to the global model's i^{th} layer, $d^{i_p} = \|w_t^i - w_{t+1}^{i_p}\|$ is calculated as in line 10. To calculate L^2 norm distances, weight matrices are converted 1D vectors.

The lower thresholds and the upper thresholds for the safety region, $[Q_1^i - 1.5 \times IQR^i, Q_3^i + 1.5 \times IQR^i]$ are calculated for each layer based on the boxplot outlier

[1] Unless otherwise stated, all distances in this study refer to the Euclidean distance.

elimination method as in lines 11 & 12. The $RS(d^{i_p})$ shows the **R**ealibility **S**tatus of the distance of i^{th} layer of a collaborator p. Suppose a collaborator stays in the safety region and is not an outlier for any layer of the model architecture. In that case, the collaborator is labeled as reliable and is eligible to participate in global model aggregation. In other words, if $RS(d^{1_p}) \wedge RS(d^{2_p}) \wedge ... \wedge RS(d^{I_p}) = 1$ (is $True$), the collaborator is labeled as reliable $w_{t+1}^{p_{rel}}$ through lines 14-16. Finally, the weighted average of reliable collaborators' parameters are calculated as the new parameters of the global for the next FL round based on their relative sample size in line 18.

Algorithm 2 BARFED. T is the number of communication rounds, P is the number of collaborators, s is the relative sample size, i is used to index layers where I is the total number of layers, and J is the number of local epochs. Q_1^i, Q_3^i, and IQR^i are first quartile, third quartile and interquartile range for the layer i, respectively.

```
 1: procedure ServerUpdate
 2:     Server initializes weights
 3:     for each round $t = 1, 2, ...T$ do
 4:         send $w_t$ (main model weights) to the collaborators
 5:         for each collaborator $p = 1, 2, ...P$ do in parallel
 6:             $w_{t+1}^{p} \leftarrow$ CollaboratorUpdate($p, w_t$)        ▷ Local Training
 7:         end for
 8:         for each layer of the model $i = 1, 2, ...I$ do
 9:             for each collaborator $p = 1, 2, ...P$ do
10:                 $d^{i_p} \leftarrow \|w_t^i - w_{t+1}^{i_p}\|$  ▷ Calculate the distance of each collaborator
11:             $lower_{thr}^{i} \leftarrow Q_1^i - (1.5 \times IQR^i)$      ▷ the lower bound for each layer
12:             $upper_{thr}^{i} \leftarrow Q_3^i + (1.5 \times IQR^i)$      ▷ the upper bound for each layer
13:
14:         for each collaborator $p = 1, 2, ...P$ do
15:             if $lower_{thr}^{i} < d^{i_p} < upper_{th}^{i}, \forall i$ then
16:                 mark collaborator $\boldsymbol{p}$ as reliable, $p_{rel}$
17:
18:         $w_{t+1} \leftarrow \sum_{n=1}^{n_{rel}} s_{p_{rel}} \times w_{t+1}^{p_{rel}}$        ▷ FedAvg with reliable collaborators
19:
20: procedure CollaboratorUpdate($p, w$)
21:     for each step $j = 1, 2, ...J$ do
22:         $w_{t+1}^{p} \leftarrow w_t^p - \eta \nabla \ell(x^p, y^p; w)$
23:     return $w$ to the server
```

In my proposed method, even when a collaborator is an outlier according to a single layer, the collaborator is labeled as malicious and removed from the aggregation calculation. This approach is one of the main differences between BARFED and other distance-based defense strategies. Most of the defense strategies reviewed in the literature partially incorporate local model parameters to the aggregation rule by evaluating each parameter in the model separately. For this reason, some parameters of a collaborator can be excluded from the aggregation, while other parameters of the same collaborator can be considered in the updated global model. My method evaluates each collaborator holistically with an all-or-nothing approach. Since model parameters are highly dependent on each other in neural network architectures, evaluating them independently from each other can lead to faulty inferences. If a collaborator is thought to be malicious in any layer, it indicates that the collaborator is malicious, and the reliability of its entire model is questionable. For this reason, the collaborator completely discarded from the aggregation. In other words, consensus should be ensured among all layers of a collaborator's model to determine that the collaborator is a reliable one and its model is credible.

Lemma 1. *The expected time complexity of the BARFED*$(V_1, V_2, ..., V_n)$ *is* $\mathcal{O}(l \cdot n \cdot (d + \log n))$ *where* $V_1, V_2, ..., V_n$ *has* l *layer and each layer is* d*-dimensional vector.*

Proof. For each layer in V_i, server computes n distances for d-dimensional vectors ($\mathcal{O}(nd)$). Then, server finds the outliers by sorting the distances of each collaborator, ($\mathcal{O}(n \log n)$). Finally, each layer is checked if it is outlier or not ($\mathcal{O}(n)$). As a result, a time complexity of $\mathcal{O}(l \cdot (nd + n \log n + n))$ is obtained. Hence, the time complexity can be reduced to $\mathcal{O}(l \cdot n \log n)$. □

The computational complexity of the BARFED is much more efficient than Krum and its variant Bulyan (see Section 3.2), which are quadratic ($\mathcal{O}(d \cdot n^2)$). BARFED is slightly more efficient than the coordinate-wise median and trimmed mean as they require sorting of all individual parameters (BARFED only makes sorting as many as the number of layers).

CHAPTER 5

EXPERIMENTAL DESIGN

In order to evaluate the defense performance of BARFED and compare it with the performances of other methods in the literature, I have carried out extensive experiments. The experimental design details are given in this chapter, and the results are presented in the next chapter. The datasets used in the experiments are introduced in Section 5.1. The details of FL settings are explained in Section 5.2. The types of attacks considered are given in Section 5.3. The attacker types are explained in Section 5.4. The baseline methods to compare with BARFED are introduced in Section 5.5. Lastly, the model architecture and hyperparameters used in the experiments are given in Section 5.6.

5.1 Datasets

The experiments have been conducted on three different data sets: MNIST [86], CIFAR10 [87], and Fashion-MNIST [88] datasets.

MNIST dataset contains 28×28 grayscale handwritten digit images as exemplified in Figure 6. The MNIST data set consists of 50.000 training, 10.000 validation, and 10.000 testing images. For MNIST, the number of images with distinct labels are given in Table 1

Table 1: The number of each label in each sets.

Label	Train	Valid	Test	Total
0	4932	991	980	6903
1	5678	1064	1135	7877
2	4968	990	1032	6990
3	5101	1030	1010	7141
4	4859	983	982	6824
5	4506	915	892	6313
6	4951	967	958	6876
7	5175	1090	1028	7293
8	4842	1009	974	6825
9	4988	961	1009	6958
Total	**50000**	**10000**	**10000**	**70000**

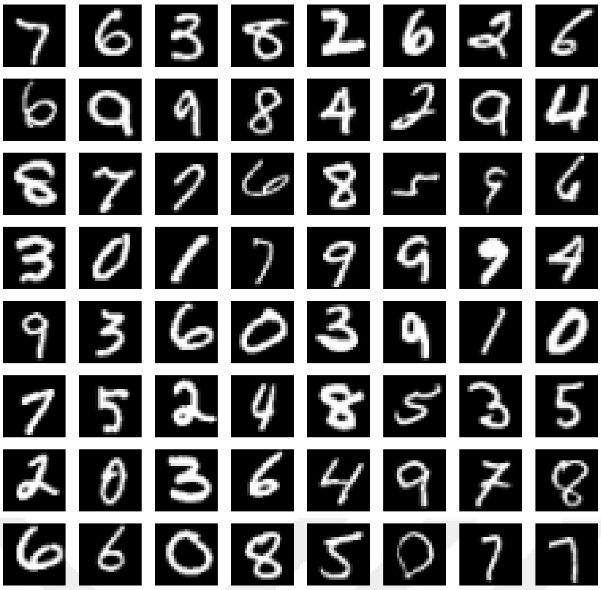

Figure 6: MNIST dataset .

CIFAR10 is an image dataset containing 32×32 color images of 10 different classes: plane, car, bird, cat, deer, dog, frog, horse, ship, and truck. CIFAR10 consists of 50.000 training and 10.000 testing images with an equal number of images for each class. Sample images from the data can be seen in Figure 7.

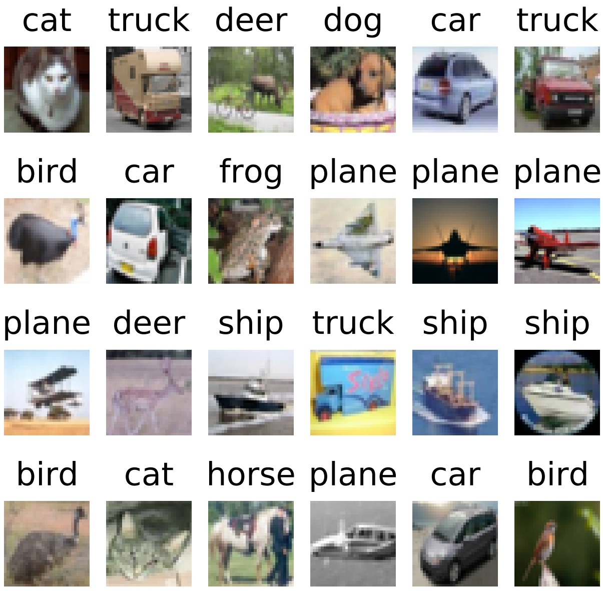

Figure 7: Example demonstration of CIFAR10 dataset .

The fashion-MNIST dataset contains 28×28 grayscale images of 10 classes with 60,000 training images and 10,000 testing images. The image classes are T-shirt/Top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle Boot (See Figure 8).

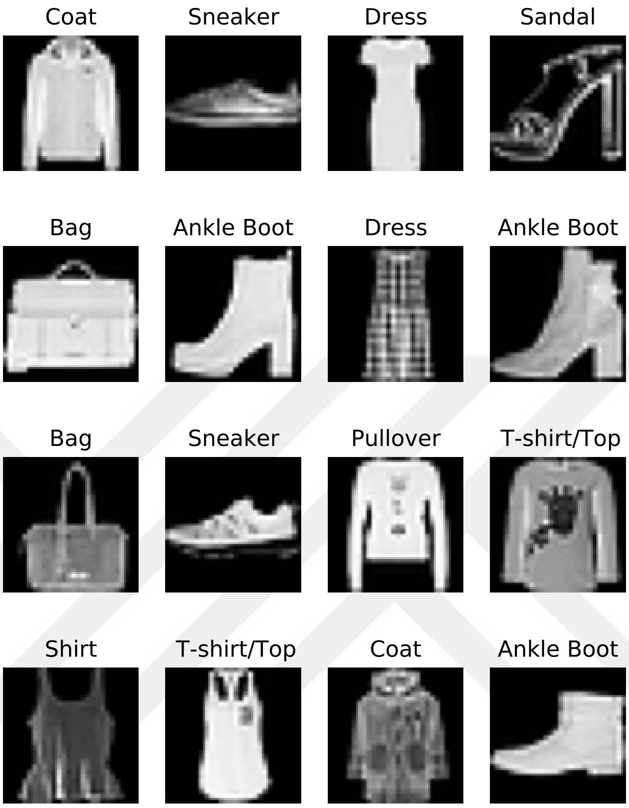

Figure 8: A sample from Fashion MNIST dataset.

5.2 FL Settings

In an FL setting, two issues are important: collaborators and the distribution of the collaborators' data. There are two possible data distribution cases: IID (independent identically distributed) or Non-IID. In the experiments, I mainly consider the distribution of label classes over collaborators.

In the IID setting, data is randomly and uniformly distributed to the collaborators. That is, each collaborator has an equal number of examples of each class. In CIFAR 10 and Fashion MNIST cases, each collaborator has each class equally (See Table 1). A small adjustment was made for the MNIST dataset, and an equal number of images from each class were included in the MNIST experiments by preserving the maximum number of data. For example, to see the performance of the global model on each class equally, 890 images are sampled from each class for the test set (see Table 1). The selection was made randomly after shuffling the data.

In the non-IID setting, each participant has examples of two randomly selected classes for MNIST and Fashion-MNIST cases and examples of five randomly selected classes for CIFAR 10 cases.

5.3 Attack Types

In the experiments, I evaluated the performances of several FL defense algorithms in no attack case and under label flipping and Byzantine attack cases with various malicious collaborator ratios.

Label flipping attacks are examples of data poisoning. Malicious collaborators replace the ground-truth label of the training data with a target class label in the label flipping attacks. For example, a malicious collaborator with label 7 in the dataset replaces the label with 1 and then performs local training to mislead the FL process.

Byzantine attacks are examples of model poisoning. In Byzantine attacks, arbitrary parameter updates are sent from malicious collaborators. In the experiments, malicious collaborators send random weight updates drawn from the standard normal distribution with zero mean and unit standard deviation.

5.4 Attacker Types

In the experiments, I have considered two kinds of attackers: Independent and Organized.

Independent attackers are malicious collaborators who cannot coordinate with each other, act individually, and send distorted information to the server in a disengaged manner. On the other hand, organized attackers are capable of coordinating their activities with the other attackers, and they usually send the same or similar updates. Therefore, organized attackers can also be called coordinated attackers.

To illustrate, in independent label flipping attacks, malicious collaborators replace their ground truth labels with arbitrary target labels. Considering the two malicious collaborators with label 7 in their dataset, one replaces its ground truth label with label 1 while the other malicious collaborator replaces with label 4. This is an example of an independent attack. On the other hand, all malicious collaborators replace their ground labels with consistent target labels in organized label flipping attacks. Considering the previous example, all malicious collaborators may replace label 7 with label 1.

In a similar way, malicious collaborators send different random weight updates in independent Byzantine attacks while they send the same random weights in the organized Byzantine attacks.

In the experiments, my aim was to make the attacks more successful, reduce the likelihood of malicious collaborators being detected, and finally affect the performance of the main model more severely. Hence, in label flipping attacks, the replaced classes are chosen as semantically similar as possible. The replaced classes in the organized setting for each data set are presented in Table 2.

Table 2: Replaced classes for organized label flipping attack.

MNIST		**Fashion-MNIST**		**CIFAR10**	
Original	Replaced	Original	Replaced	Original	Replaced
0	9	T-shirt/Top	Shirt	Plane	Bird
1	7	Trouser	Dress	Car	Truck
2	5	Pullover	Coat	Bird	Plane
3	8	Dress	Trouser	Cat	Dog
4	6	Coat	Pullover	Deer	Horse
5	2	Sandal	Sneaker	Dog	Cat
6	4	Shirt	T-shirt/Top	Frog	Ship
7	1	Sneaker	Ankle Boot	Horse	Deer
8	3	Bag	Sandal	Ship	Frog
9	0	Ankle Boot	Sneaker	Truck	Car

5.5 Baseline Method Selection

In order to compare the performance of BARFED with other defense mechanisms, I have chosen a set of baseline algorithms that are compatible with my FL settings.

The first baseline algorithm that I evaluate is "NoDefense" which is identical to the vanilla FedAvg algorithm. I included this algorithm because I want to gauge the impacts of various attacks on the learning performance and to evaluate the defense provided by BARFED and other baseline defense mechanisms.

The second baseline algorithm is "Coordinate-Wise Median" [22], or simply "CwMedian", which can be seen as the simplest variant of the FedAvg learning algorithm. Basically, instead of the weighted mean operator used in FedAvg, this algorithm uses the median operator to aggregate the updates received from collaborators. Therefore, this algorithm can also be regarded as a defense mechanism as it tends to discard the updates received from malicious collaborators that are usually far away from the other collaborators' updates. Like BARFED, this algorithm is an assumption-free algorithm.

The third baseline algorithm employed is "TrimmedMean" [22] that discards the updates with extreme values before applying the weighted mean operator in the aggregation step of FedAvg. In this method, it is necessary to determine a parameter that defines the fraction of extreme values that will be discarded. Ideally, the value of this parameter should be set to the malicious collaborator ratio in the system. Obviously, it is unrealistic to know this ratio in a practical FL setting.

5.6 Model Architectures and Hyper-parameters

This section introduces used model architectures and hyperparameters for each dataset.

5.6.1 Model Architectures

The model architectures used for each datasets are shown in Table 3 (MNIST-2NN [2]), Table 4 (MNIST-CNN [2]), Table 5 (CIFAR10 [83]) and Table 6 (Fashion-MNIST). I used ReLU activation function in all models.

Table 3: Model architecture of MNIST-2NN.

Layer	**Size**
Fully Connected	(784, 200)
Fully Connected	(200, 200)
Fully Connected	(200, 10)

Table 4: Model architecture of MNIST-CNN.

Layer	**Size**
Conv	32@5×5
Max Pooling	2×2
Conv	64@5×5
Max Pooling	2×2
Fully Connected	(1024, 512)
Fully Connected	(512, 10)

Table 5: Model architecture of CIFAR10.

Layer	**Size**
Conv	32@3×3, pad=1
Conv	32@3×3, pad=1
Max Pooling	2×2
Conv	64@3×3, pad=1
Conv	64@3×3, pad=1
Max Pooling	2×2
Conv	128@3×3, pad=1
Conv	128@3×3, pad=1
Max Pooling	2×2
Fully Connected	(2048, 128)
Fully Connected	(128, 10)

Table 6: Model architecture of Fashion-MNIST.

Layer	Size
Conv	32@5×5, pad=2
Max Pooling	2×2
Conv	64@5×5, pad=2
Max Pooling	2×2
Fully Connected	(3136, 500)
Fully Connected	(500, 10)

5.6.2 Hyperparameters

The FL setting parameters used for each dataset are shown in Table 7. Learning rate scheduling, gradient clipping and data augmentation techniques such as random horizontal flip and random clipping have only been applied for the CIFAR10 dataset. Therefore, related parameters are marked with N/A (Not Applicable) for MNIST and Fashion-MNIST.

Table 7: FL setting parameters used in experiments.

Parameters	**MNIST-2NN**	**MNIST-CNN**	**Fashion MNIST**	**CIFAR10**
number of collaborators (n)	100	100	100	100
communication round (t)	200	200	200	500
number of label in each collaborator in IID setting	2	2	2	5
number of label in each collaborator in Non-IID setting	10	10	10	10
batch size	32	32	25	100
number of epoch	10	10	10	10
momentum	0.9	0.9	0.9	0.9
learning rate	0.01	0.01	0.002	0.0015
minimum learning rate (min_lr)	N/A	N/A	N/A	0.000010
lr scheduler factor	N/A	N/A	N/A	0.2
best threshold	N/A	N/A	N/A	0.0001
clipping threshold	N/A	N/A	N/A	10

CHAPTER 6

EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, the performance of BARFED is evaluated and compared with the performances of prominent defense mechanisms that I call baseline methods proposed in the literature. Any defense mechanism should not deteriorate the performance of a learning algorithm under no attack cases, as well as mitigating the performance loss caused by attacks. Hence, first of all, I evaluate the no attack case in Section 6.1. Then, the performances of BARFED and the selected baseline algorithms are evaluated with MNIST, Fashion-MNIST, and CIFAR10 datasets in the subsequent Section 6.2, Section 6.3, and Section 6.4. Lastly, a general discussion is presented in Section 6.5.

In this chapter, I evaluate four different classification models trained on three different datasets. The performance metric that I consider in performance evaluations is classification accuracy which is equal to the rate of correct classifications on independent test sets. The attacks aim to compromise both the performance and convergence of the models, thereby causing oscillations in accuracy scores. Due to such oscillations, it would be misleading to report only the accuracy obtained at the last FL round. Moreover, as the highest or the lowest points of oscillations may occur randomly, it is also challenging to compare different algorithms objectively. Therefore, in the experiment results, I report both the lowest and highest scores of the last ten FL rounds to reveal the algorithm performance and also the severity of such oscillations.

The experiments carried out are essentially multi-class classification tasks. Hence, f-macro and f-micro metrics could also be used to evaluate results. However, in the experimental setup, the test sets contain each label equally (i.e., the classes are balanced). For this reason, I preferred to use accuracy in comparisons.

6.1 No Attack

It is usual to test the performance of a defense methodology on different attack scenarios. However, in a practical setting, I cannot know when the attack will occur. Hence, defense mechanisms are always actively used regardless of whether there is an attack or not. For this reason, the defense mechanism integrated into the FL system should not cause non-negligible performance loss also in the absence of an attack. In other words, ideally, the defense to be integrated into the system should work well even when there is no attack. In such cases, the method can be called a solid defensive strategy.

Table 8 and Figure 9 show the results of experiments in IID case without any attack, i.e., all collaborators are trusted and there is no malicious collaborator.

It is important to note that when there is no attack in the system, the TrimmedMean method gives the same result as the NoDefense case because no collaborator updates are discarded from the weighted mean calculation.

BARFED, TrimmedMean, and the CwMedian show similar performances in the IID setting and do not cause any significant performance degradation. Although the accuracy scores on the test sets obtained from Fashion MNIST and CIFAR for CwMedian are marginally worse, the difference can be seen as negligible if I consider practical FL characteristics and settings.

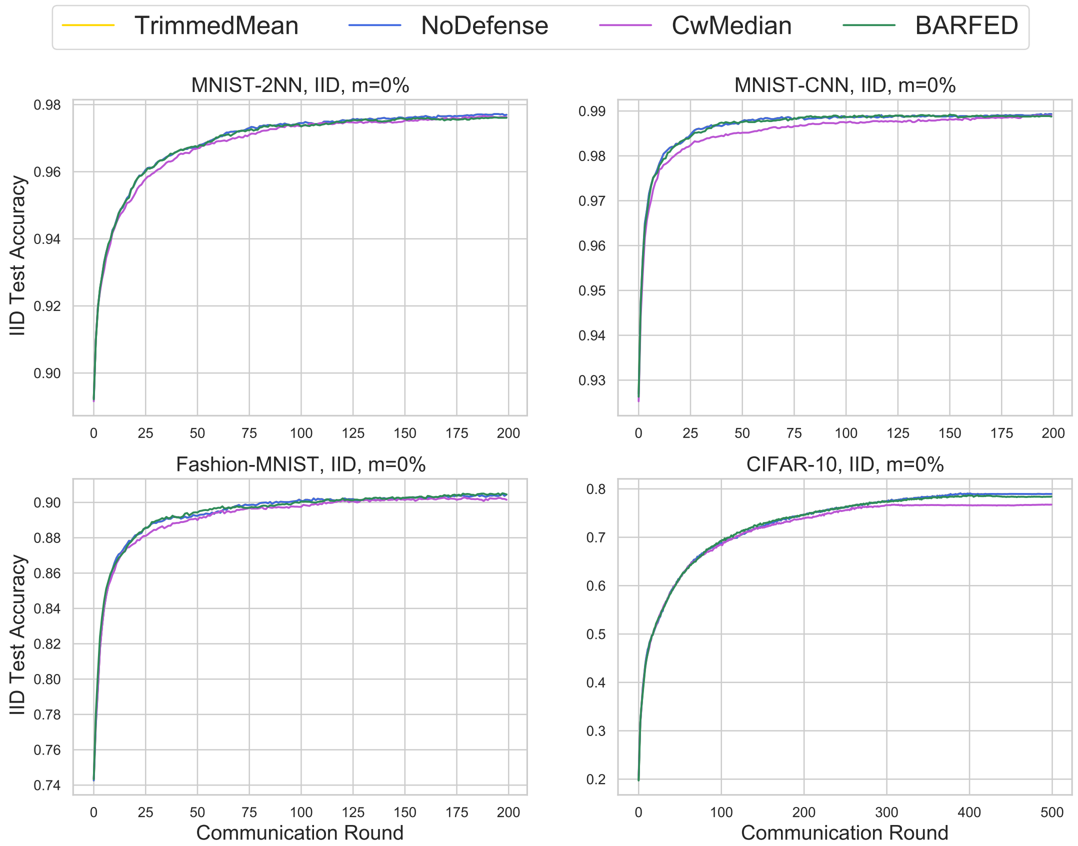

Figure 9: Accuracy curves of different strategies when all collaborators are trusted (i.e., $m = 0\%$) in the IID case.

On the other hand, my proposed method outperforms the others in the Non-IID experimental setting. Table 9 and Figure 10 show the results of experiments when all collaborators are trusted in the non-IID case. As it can be seen from the results, CwMedian is the worst algorithm in all cases. This is mainly due to the non-IID

Table 8: Accuracy scores on test sets when all collaborators are trusted (i.e., $m = 0\%$) in the IID case. The best results are bold.

	MNIST-2NN		**MNIST-CNN**		**Fashion MNIST**		**CIFAR10**	
	min	max	min	max	min	max	min	max
NoDefense	**97.7**	**97.7**	**98.9**	**98.9**	**90.4**	**90.5**	**78.9**	**79.0**
TrimmedMean[1]	**97.7**	**97.7**	**98.9**	**98.9**	**90.4**	**90.5**	**78.9**	**79.0**
CwMedian	97.6	97.6	**98.9**	**98.9**	90.1	90.2	76.7	76.8
BARFED	97.6	97.6	**98.9**	**98.9**	**90.4**	**90.5**	78.4	78.4

[1] The same results as NoDefense

training data employed in these experiments. That is, the median operator, which returns one of the participants' model parameters as the global model's parameter, seems not to produce a representative model. Except for the Fashion-MNIST case, BARFED and FedAvg (NoDefense and TrimmedMean are equivalent to FedAvg in no attack case) perform similarly. In Fashion-MNIST, the collaborator models are likely to be very different due to the non-IID training data in collaborators. Hence, BARFED potentially discards many participants in the aggregation step.

Table 9: Accuracy scores on test sets when all collaborators are trusted (i.e., $m = 0\%$) in Non-IID case. The worst results are bold.

	MNIST-2NN		**MNIST-CNN**		**Fashion MNIST**		**CIFAR10**	
	min	max	min	max	min	max	min	max
NoDefense	96.4	96.6	98.8	98.8	86.8	87.4	77.5	77.6
TrimmedMean[1]	96.4	96.6	98.8	98.8	86.8	87.4	77.5	77.6
CwMedian	**80.7**	**85.3**	**96.9**	**97.1**	**79.1**	**80.0**	**64.3**	**64.6**
BARFED	96.2	96.4	98.7	98.8	82.4	84.3	77.8	77.9

[1] The same results as NoDefense

As these experiments show, BARFED and other evaluated approaches do not introduce a significant performance penalty in no attack case for IID training data in collaborators. On the other hand, non-IID training data is challenging for all defense methods considered. Nevertheless, the performance loss caused by BARFED is plausible and marginal in the majority of cases evaluated.

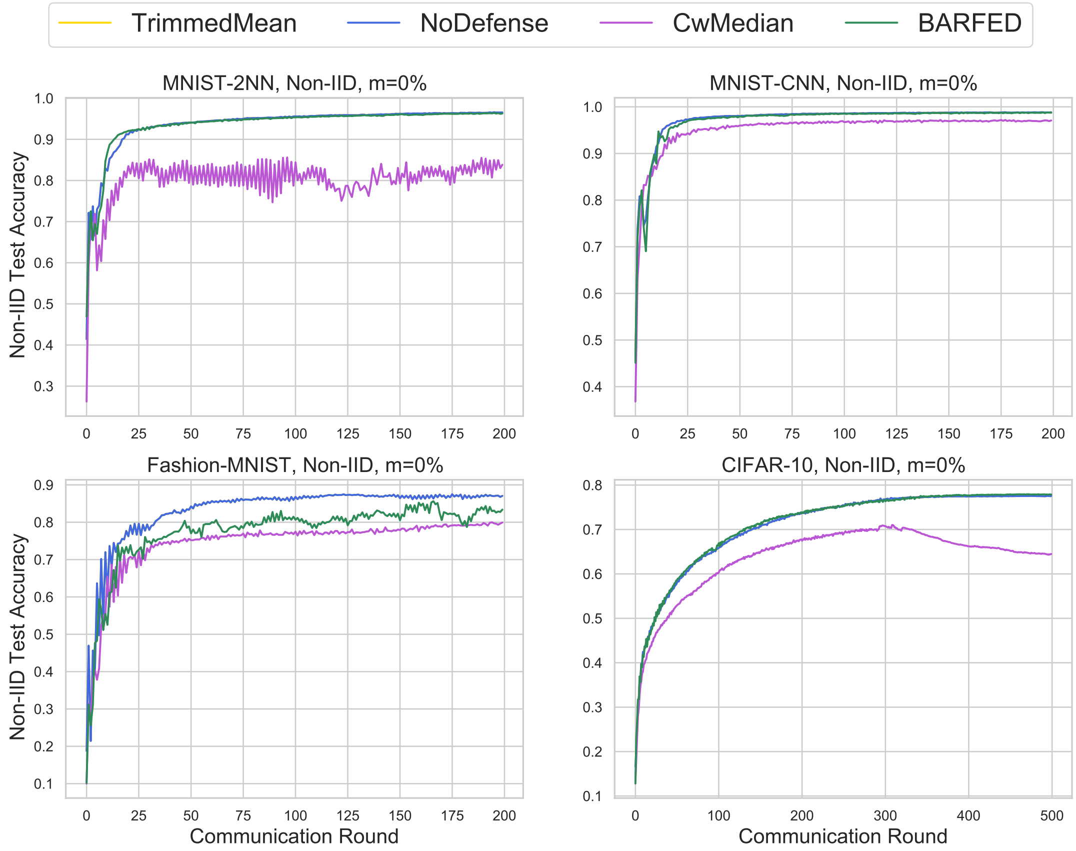

Figure 10: Accuracy curves of different strategies when all collaborators are trusted (i.e., $m = 0\%$) in the Non-IID case.

6.2 MNIST

For MNIST dataset, I have trained two kinds of models: 2NN (see Table 3) and CNN (see Table 4). In this section, the performances of the defense mechanisms applied to these models are evaluated for various kinds of attacks with various malicious collaborator ratios.

6.2.1 MNIST 2NN

6.2.1.1 Label Flipping Attacks

Table 10 and Figure 11 present the results of the experiments carried out for label flipping attack on MNIST dataset with 2NN model when the collaborators' data are IID.

The maximum performance under no attack case for this model is 97.7 (see Table 8). When the malicious collaborator ratio (m) is 10% or 20%, neither organized nor independent attacks cause significant performance losses as all methods including NoDefense, achieve maximum accuracy values that are close to the maximum accuracy of FedAvg in no attack case. Nevertheless, a small performance loss can be observed when the malicious collaborator ratio increases. Although the differences are marginal, BARFED has provided the best performance in all these cases.

Another important point to note is that for $m = 10\%$, attacks do not even have a significant effect on the evolution of the algorithms across FL rounds, as evidenced by very close minimum and maximum accuracies obtained with all algorithms. Therefore, I can say that an attack is ineffective when attackers are independent, and m is small. On the other hand, organized attacks cause large oscillations in accuracy values, and the gap between the minimum and maximum accuracies obtained gets larger in the NoDefense case. When $m = 20\%$, even independent attacks can cause such oscillations. According to these results, I can say that BARFED and other defense algorithms considered are capable of mitigating label flipping attacks successfully in all evaluated IID cases. The detailed performance comparisons of methods can be found in the Appendix A.

Table 11 and Figure 12 present the results of the experiments carried out for label flipping attack on MNIST dataset with 2NN model when the collaborators' data are non-IID.

Similar to the IID case, as the malicious collaborator ratio increases, the performance loss increases, and organized attacks become more effective than independent attacks. If two data distribution settings are compared, the performance loss caused by attacks in the non-IID case is greater than that of the IID case. In the non-IID setting, there are significant differences between the minimum and the maximum accuracies achieved

Table 10: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in IID setting for MNIST-2NN architecture. The best results are bold.

	Organized				Independent			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	92.9	97.6	72.4	97.3	97.0	97.2	91.3	97.1
CwMedian	97.4	97.4	97.2	97.3	97.4	97.5	97.0	97.1
TrimmedMean	97.5	97.5	96.9	97.0	97.5	97.5	97.1	97.1
BARFED	**97.6**	**97.7**	**97.4**	**97.5**	**97.6**	**97.6**	**97.4**	**97.5**

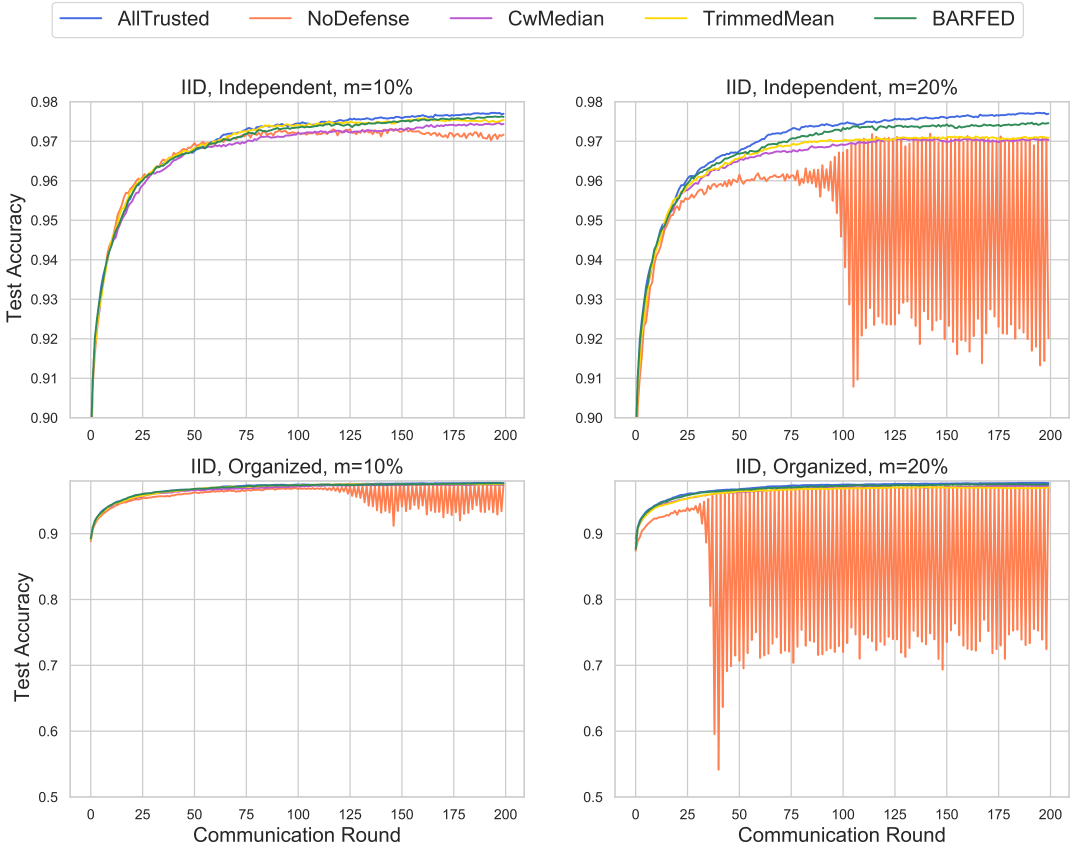

Figure 11: Accuracy curves for MNIST-2NN under label flipping attacks at different attacker ratios for the IID case.

by NoDefense regardless of attack types and malicious collaborator ratios. Hence, it is possible to say that the non-IID case is worse than the IID case for all attack scenarios.

As it can be seen from Table 11 and Figure 12, CwMedian could not cope with the attack and worsened the situation for all attack scenarios in the non-IID case. The maximum accuracy drops down to 75.0% when attackers are organized and $m = 20\%$. Moreover, the accuracies provided by CwMedian are even worse than the accuracies obtained by NoDefense. TrimmedMean can recover the performance loss when $m = 10\%$, however it gets worse scores than FedAvg when $m = 20\%$, i.e., the performance score drops down to 79.9%. BARFED is able to defend against the harmful effects of the attack successfully and gives the highest accuracy scores among all defense methods. Moreover, its accuracy is close to the case in which there are no malicious collaborators (i.e., no attack case).

Table 11: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in the non-IID setting for MNIST-2NN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	92.4	95.8	83.8	88.3	93.7	95.3	89.3	94.0
CwMedian	75.1	83.8	67.7	75.0	80.8	83.3	67.8	75.9
TrimmedMean	94.6	95.5	79.9	87.7	95.1	95.4	80.9	89.7
BARFED	**96.1**	**96.3**	**95.6**	**96.1**	**96.0**	**96.3**	**95.5**	**96.1**

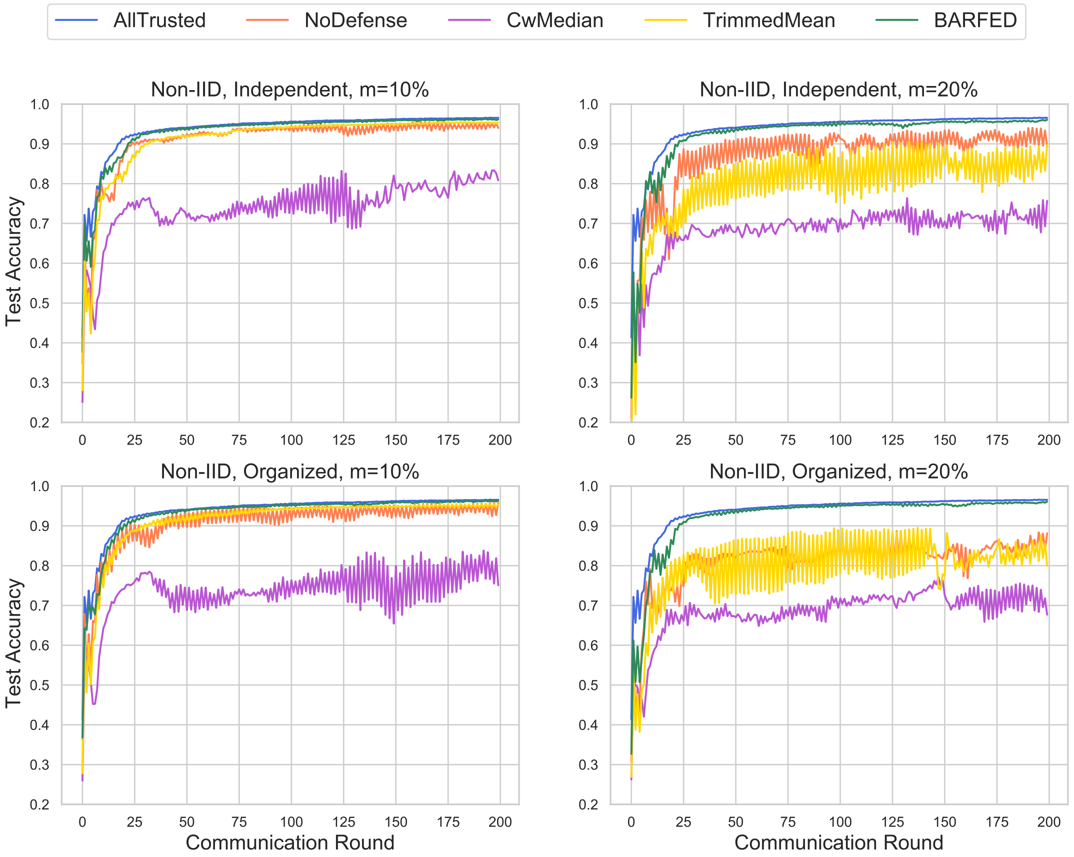

Figure 12: Accuracy curves for MNIST-2NN under label flipping attacks at different attacker ratios in the non-IID case.

6.2.1.2 Byzantine Attacks

Table 12 and Figure 13 show the results of the Byzantine attack scenarios for MNIST 2NN architecture when the collaborators' data are IID.

Similar to label flipping attacks, organized attackers cause more performance degradation. The worst accuracy score is recorded when the attackers are organized and $m = 20$. In this case, test accuracies in the last 10 FL rounds oscillate between 31.9% and 43.9%. It can be said that, without any defense mechanism, Byzantine attacks result in more performance loss compared to label flipping attacks. On the other hand, all defense methods considered perform well in IID cases with independent or organized Byzantine attacks. Although the performances of all defense methods are very close to each other, BARFED provides the best resistance among all methods.

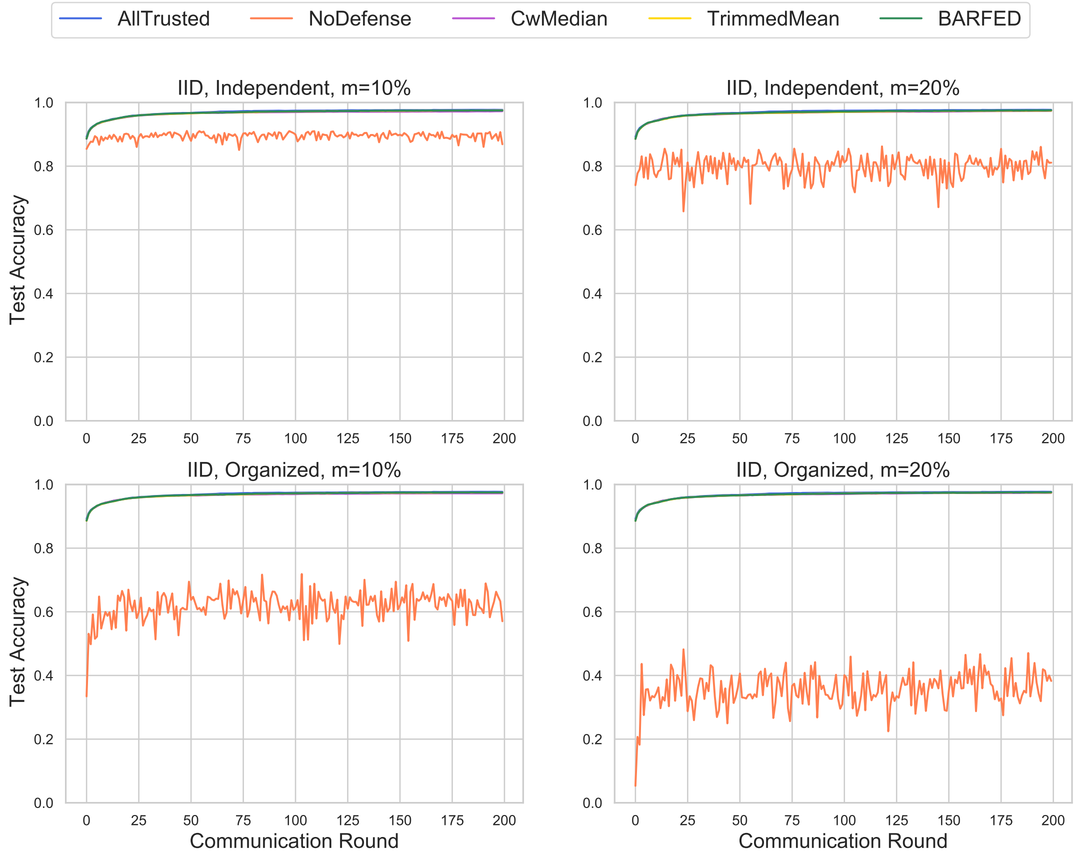

Figure 13: Accuracy curves of MNIST-2NN under Byzantine attacks at different attacker ratios for the IID case.

Table 12: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in the IID setting for MNIST-2NN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	57.0	68.9	31.9	43.9	86.0	90.7	76.2	86.1
CwMedian	97.2	97.2	97.3	97.4	97.2	97.3	97.3	97.3
TrimmedMean	97.4	**97.5**	97.4	97.5	**97.5**	**97.5**	97.4	97.4
BARFED	**97.5**	**97.5**	**97.5**	**97.6**	**97.5**	**97.5**	**97.5**	**97.6**

Table 13 and Figure 14 show the results of the Byzantine attack scenarios for MNIST 2NN architecture when the collaborators' data are non-IID.

As the results indicate, Byzantine attacks in non-IID settings are more dangerous than such attacks in IID settings. The largest performance degradation occurs when attackers are organized and $m = 20\%$. In this case, the accuracy oscillates between 14.9% and 26.6%. In this case, CwMedian is able to recover the harmful effects of the attack up to a point, but it cannot achieve a successful defense as TrimmedMean and BARFED do. For example, CwMedian can increase the accuracy up to 89.5%, while TrimmedMean achieves 96.1% and BARFED achieves 96.2% against organized attacks with $m = 20\%$. BARFED provides the highest scores among all methods in all cases, but the differences are insignificant.

Table 13: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in the non-IID setting for MNIST-2NN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	26.1	36.0	14.9	26.6	45.6	61.6	16.5	34.1
CwMedian	85.6	89.5	90.3	92.7	83.4	88.8	89.3	90.9
TrimmedMean	95.8	96.1	95.4	95.6	95.9	96.1	94.8	95.0
BARFED	**96.1**	**96.2**	**95.9**	**96.1**	**96.1**	**96.2**	**95.9**	**96.1**

Another important observation is that TrimmedMean performs better against Byzantine attacks than label flipping attacks. This can be explained by the fact that the parameter updates sent in Byzantine attacks are extreme and take place in the tails of the distribution. Therefore malicious collaborators are more easily detected as they are usually outliers. On the other hand, TrimmedMean cannot recover all the performance degradation in label flipping attacks, as changes in the update are more likely to be moderate.

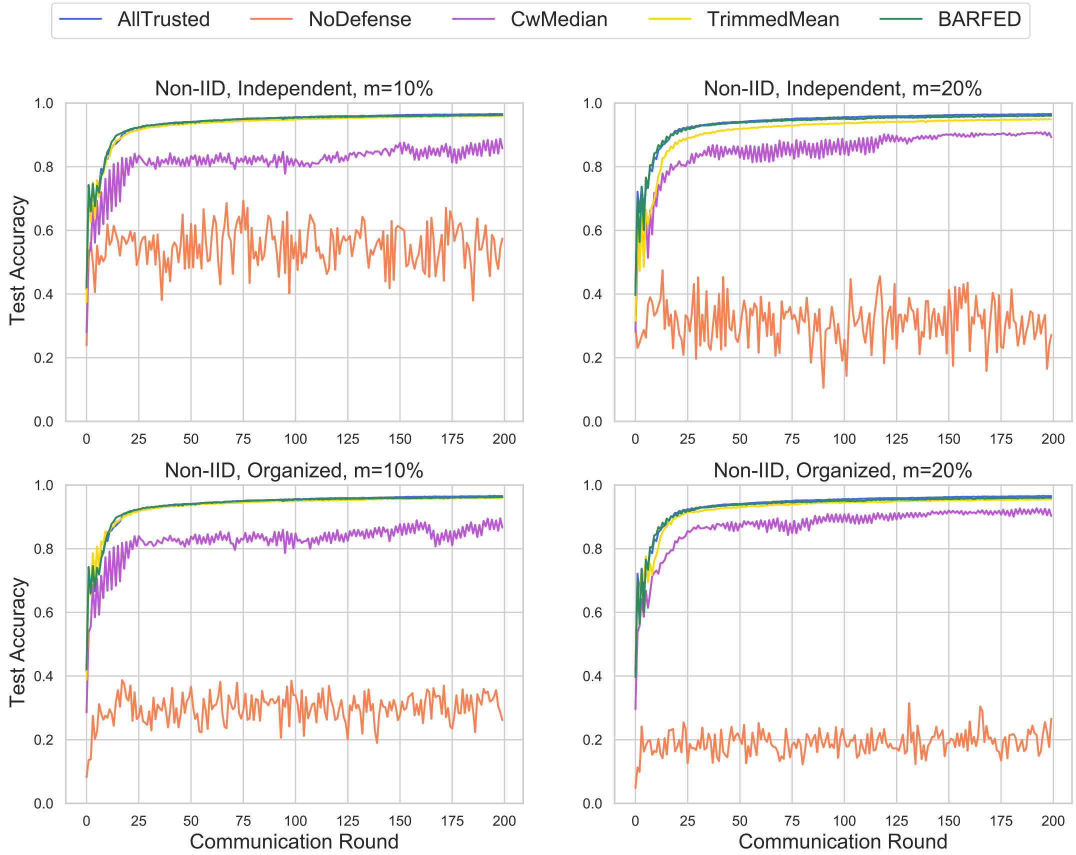

Figure 14: Accuracy curves of MNIST-2NN under Byzantine attacks at different attacker ratios for the IID case.

6.2.2 MNIST CNN

6.2.2.1 Label Flipping Attacks

Table 14 and Figure 15 present the results of the experiments carried out for label flipping attack scenarios for MNIST CNN architecture when the collaborators' data are IID.

Table 15 and Figure 16 present the results of the experiments carried out for label flipping attack scenarios for MNIST CNN architecture when the collaborators' data are non-IID.

For MNIST, the results with the CNN model are very similar to the results with the 2NN model. However, accuracy losses caused by attacks on the CNN model are smaller than that of 2NN models. So, it can be said that the MNIST-CNN model architecture is less vulnerable to label flipping attacks compared to the MNIST-2NN architecture. As the ratio of malicious collaborators increases, the severity of the oscillations and the performance degradation increase for both independent and organized attacks in both IID and Non-IID settings. As the results show, all evaluated defense methods are able to improve the performance in both IID and Non-IID cases. BARFED provides the highest accuracies when the collaborators' data are non-IID. Still, the differences between the performances of TrimmedMean and BARFED are not significant.

Table 14: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in the IID setting for MNIST-CNN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	94.2	**99.0**	75.4	**99.0**	97.8	**99.0**	96.2	98.9
CwMedian	**98.9**	98.9	98.8	98.8	98.9	98.9	**98.9**	98.9
TrimmedMean	**98.9**	98.9	98.8	98.9	**99.0**	**99.0**	**98.9**	**99.0**
BARFED	**98.9**	98.9	**98.9**	98.9	98.9	98.9	**98.9**	98.9

Table 15: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in the non-IID setting for MNIST-CNN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	95.5	98.0	83.4	91.6	97.0	98.3	95.3	96.6
CwMedian	96.4	96.7	91.4	93.1	95.4	95.9	93.5	94.2
TrimmedMean	98.5	98.6	97.3	97.6	98.6	98.6	97.2	97.6
BARFED	**98.6**	**98.7**	**98.5**	**98.6**	**98.7**	**98.8**	**98.6**	**98.7**

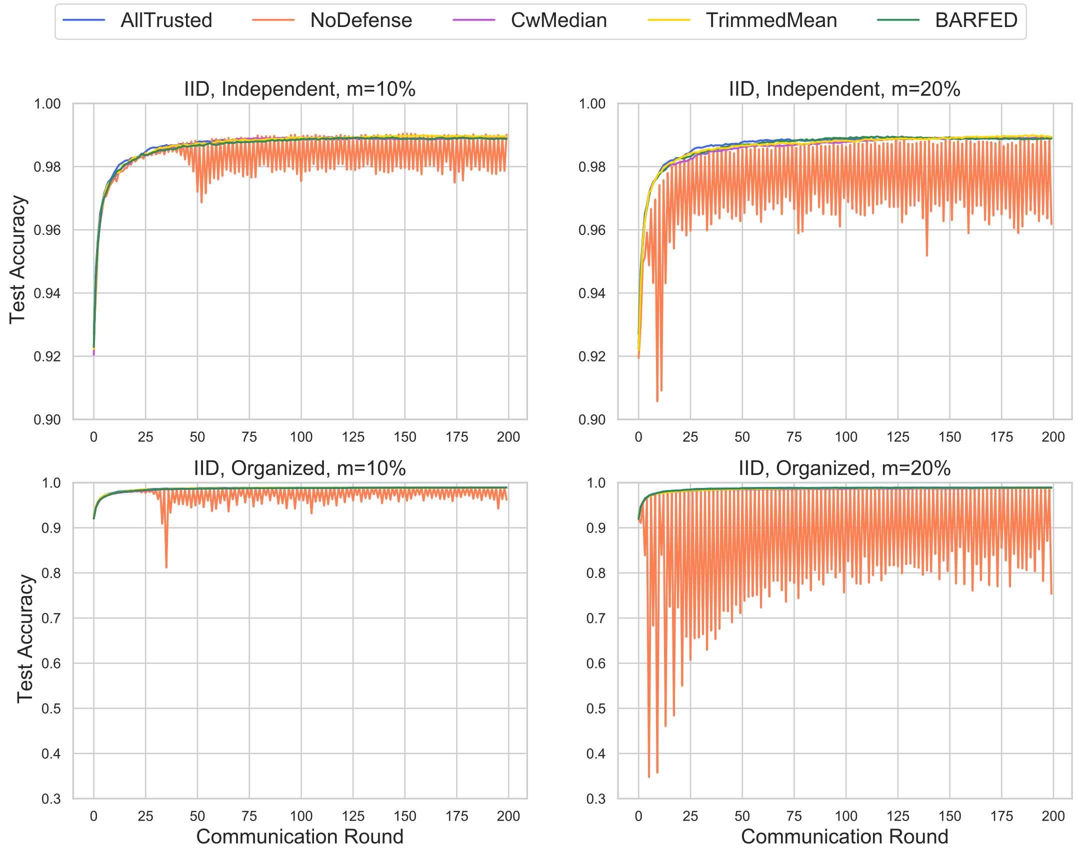

Figure 15: Accuracy curves for MNIST-CNN under label flipping attacks at different attacker ratios in the IID case.

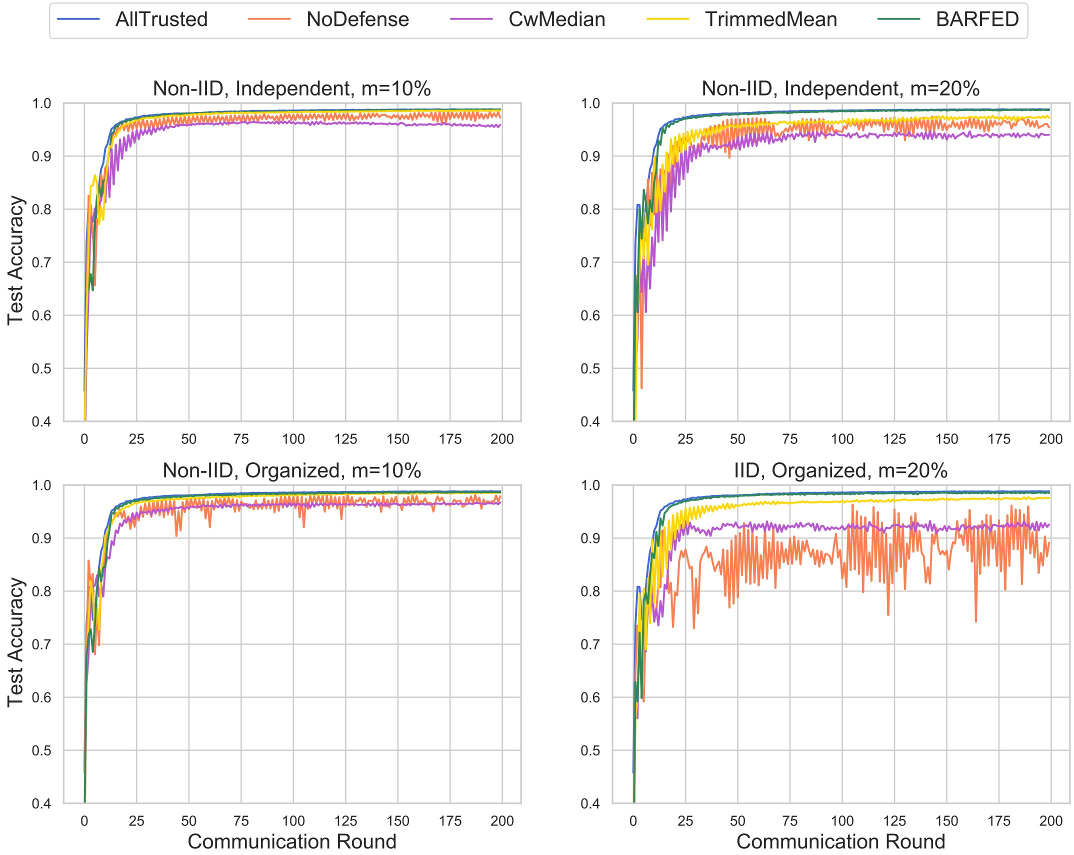

Figure 16: Accuracy curves for MNIST-CNN under label flipping attacks at different attacker ratios in the non-IID case.

6.2.2.2 Byzantine Attacks

Table 16 and Figure 17 show the results of the Byzantine attack scenarios for MNIST CNN architecture when the collaborators' data are IID.

Table 17 and Figure 18 show the results of the Byzantine attack scenarios for MNIST CNN architecture when the collaborators' data are non-IID.

If I consider CNN architecture, the performance loss is higher than 2NN architecture for Byzantine attacks. The accuracy oscillates between 10.0% and 17.0% in the IID case, while it oscillates between 9.00% and 15.7% in the non-IID case. Although the performance loss is higher for CNN architecture compared to 2NN architecture, the CwMedian was able to catch BARFED and the TrimmedMean both for IID and Non-IID cases. All methods could reverse the harmful effects of the Byzantine attacks and achieve scores as if all collaborators are trusted(no attack case).

Table 16: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in the IID setting for MNIST-CNN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	54.1	79.3	10.0	17.0	91.9	95.0	70.8	87.2
CwMedian	98.9	98.9	**98.9**	98.9	**98.9**	98.9	98.8	98.8
TrimmedMean	**99.0**	**99.0**	**98.9**	**99.0**	**98.9**	**99.0**	**98.9**	**98.9**
BARFED	98.9	98.9	98.8	98.8	**98.9**	**99.0**	98.8	98.8

Table 17: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in the non-IID setting for MNIST-CNN architecture. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	15.6	33.5	9.00	15.7	46.1	72.4	17.4	40.2
CwMedian	97.4	97.5	97.6	97.7	97.0	97.2	96.8	97.1
TrimmedMean	**98.8**	**98.8**	**98.6**	**98.7**	**98.8**	**98.8**	98.5	98.6
BARFED	98.7	**98.8**	**98.6**	**98.7**	98.7	**98.8**	**98.6**	**98.7**

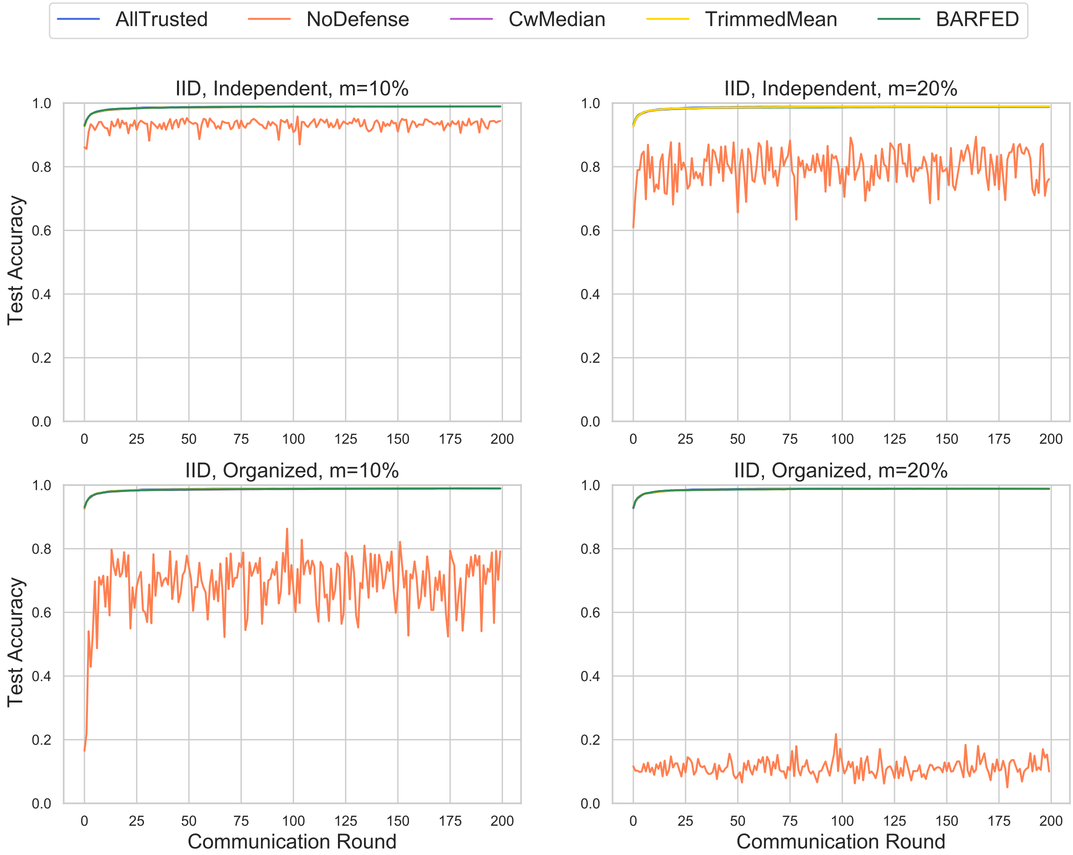

Figure 17: Accuracy curves for MNIST-CNN under Byzantine attacks at different attacker ratios in the IID case.

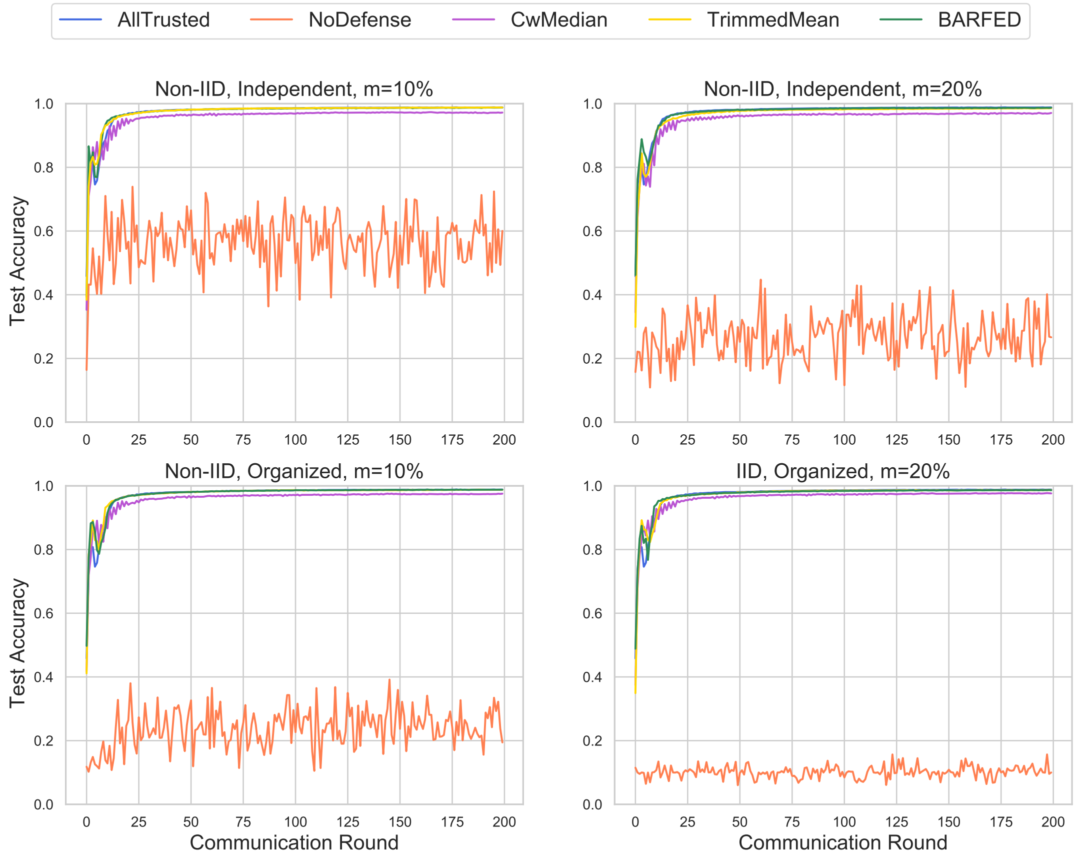

Figure 18: Accuracy curves for MNIST-CNN under Byzantine attacks at different attacker ratios in the IID case.

6.3 Fashion MNIST

6.3.1 Label Flipping Attacks

Table 18 and Figure 19 present the results of the experiments carried out for label flipping attack on Fashion-MNIST when the collaborators' data are IID.

Table 19 and Figure 20 present the results of the experiments carried out for label flipping attack on Fashion-MNIST when the collaborators' data are non-IID.

In parallel with previous experiments, organized attacks are more severe and when the collaborators' data are non-IID, the performance deteriorates. For IID experimental settings of the Fashion MNIST dataset, the attack causes extreme accuracy degradation when malicious collaborators are organized and $m\% = 20$. The accuracy degradation reaches up to 68.6% (No defense case). Recall when all collaborators are trusted, the accuracy in the IID setting is 90.4 while it is 87.4 in the non-IID setting (see Table 8, Table 9). In the same scenario for the non-IID setting, the accuracy degradation reaches up to 73.7. Interestingly, in the IID setting, the minimum accuracy obtained in the last ten FL rounds is smaller than in the Non-IID setting. On the other hand, the maximum accuracy obtained in the ten FL rounds for the non-IID setting is smaller than the IID setting. For both IID and non-IID experimental settings, BARFED outperforms and recovers the performance degradation.

Table 18: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in IID setting for Fashion-MNIST. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	87.8	89.3	68.6	88.9	89.2	89.5	83.7	89.0
CwMedian	89.8	89.9	88.6	88.7	89.6	89.7	89.2	89.3
TrimmedMean	90.0	90.1	88.8	88.9	89.9	90.0	89.0	89.2
BARFED	**90.5**	**90.7**	**90.3**	**90.4**	**90.2**	**90.3**	**90.2**	**90.3**

Table 19: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in non-IID setting for Fashion-MNIST. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	83.0	85.7	73.7	79.2	84.3	86.7	81.5	84.6
CwMedian	79.9	80.6	76.0	76.7	78.7	79.6	77.4	78.5
TrimmedMean	86.7	87.5	82.7	83.4	85.8	86.7	83.8	84.3
BARFED	**87.7**	**88.8**	**84.2**	**87.6**	**87.5**	**88.5**	**85.1**	**87.6**

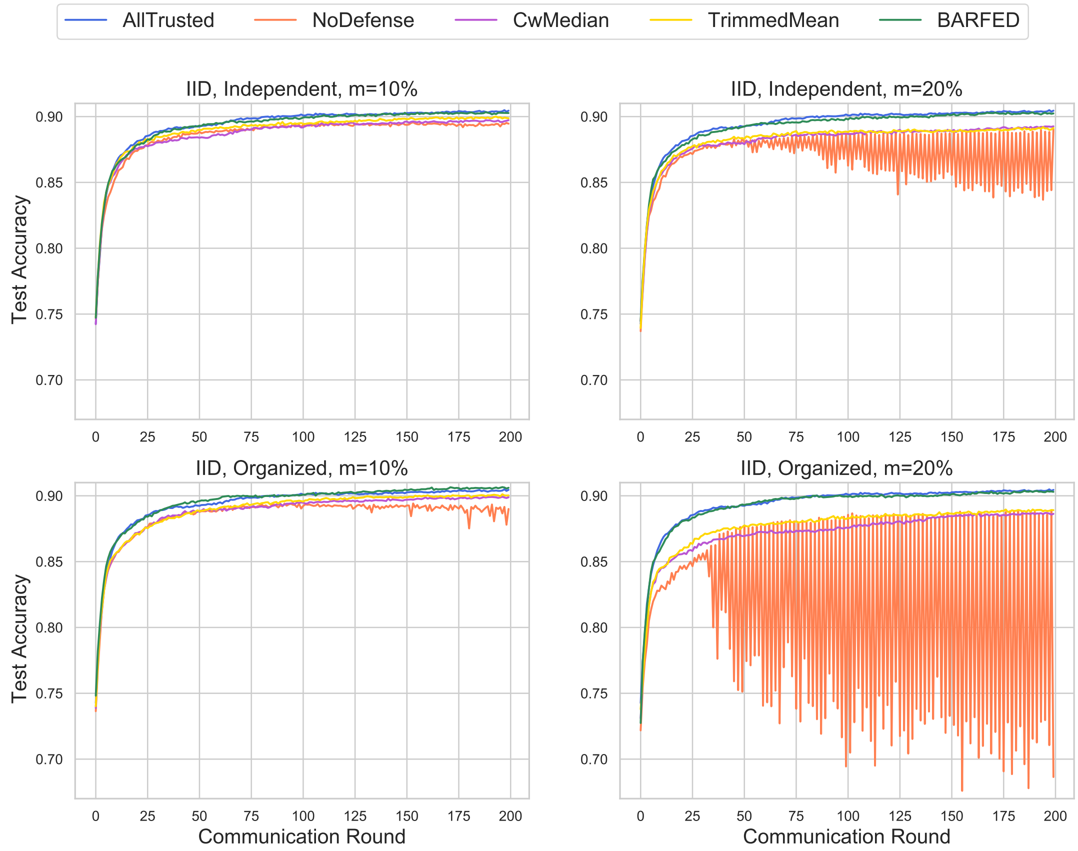

Figure 19: Accuracy curves for Fashion-MNIST under label flipping attacks at different attacker ratios in the IID case.

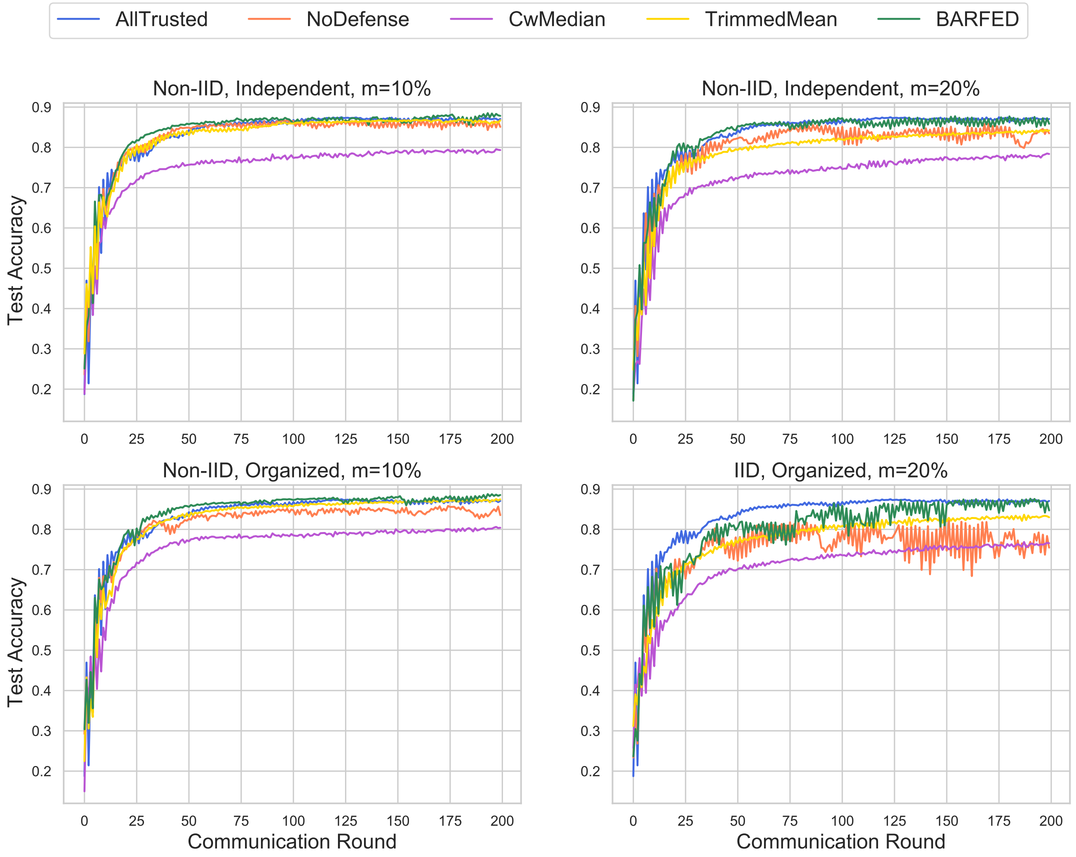

Figure 20: Accuracy curves for Fashion-MNIST under label flipping attacks at different attacker ratios in the non-IID case.

6.3.2 Byzantine Attacks

Table 20 and Figure 21 present the results of the experiments carried out for Byzantine attack on Fashion-MNIST when the collaborators' data are IID.

Table 21 and Figure 22 present the results of the experiments carried out for Byzantine attack on Fashion-MNIST when the collaborators' data are non-IID.

The effects of the attacks on the model performance are similar to the cases covered so far. Byzantine attacks cause much more performance degradation than label flipping attacks. The worst-case scenario occurs when attackers are organized and $m = 20\%$ and the accuracy scores oscillate between $9.80{=}\%$ and $21.1{=}\%$. All methods are able to recover some negative affect of the attack and increase the score for the IID case. BARFED can increase the accuracy up to 90.5%. TrimmedMean performs very similar to the BARFED, but CwMedian differs slightly in a negative direction.

Table 20: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in IID setting for Fashion-MNIST. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	15.3	48.2	9.80	21.1	65.0	79.2	36.8	60.2
CwMedian	89.9	90.1	90.1	90.2	90.0	90.2	90.0	90.1
TrimmedMean	**90.2**	**90.3**	90.2	90.3	**90.4**	**90.4**	90.2	90.3
BARFED	**90.2**	**90.3**	**90.4**	**90.5**	90.2	90.3	**90.3**	**90.4**

For Non-IID experiments with Byzantine attacks, without any defense, accuracies oscillate between 8.60% and 18.0%. CwMedian is able to eliminate the harmful effects of the attacks up to a certain point and achieve 79.6% accuracy. BARFED outperforms CwMedian and achieves an accuracy up to 86.5%. Yet, TrimmedMean gets slightly better scores than BARFED in the non-IID experimental setting.

Table 21: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in non-IID setting for Fashion-MNIST. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	5.70	23.6	8.60	18.0	19.7	39.6	10.0	22.2
CwMedian	79.4	80.8	78.4	79.6	79.7	80.8	77.7	78.3
TrimmedMean	**84.9**	**86.2**	**85.9**	**86.6**	**84.2**	**86.0**	**83.2**	84.3
BARFED	82.3	85.6	85.3	86.5	80.8	83.8	84.5	**86.3**

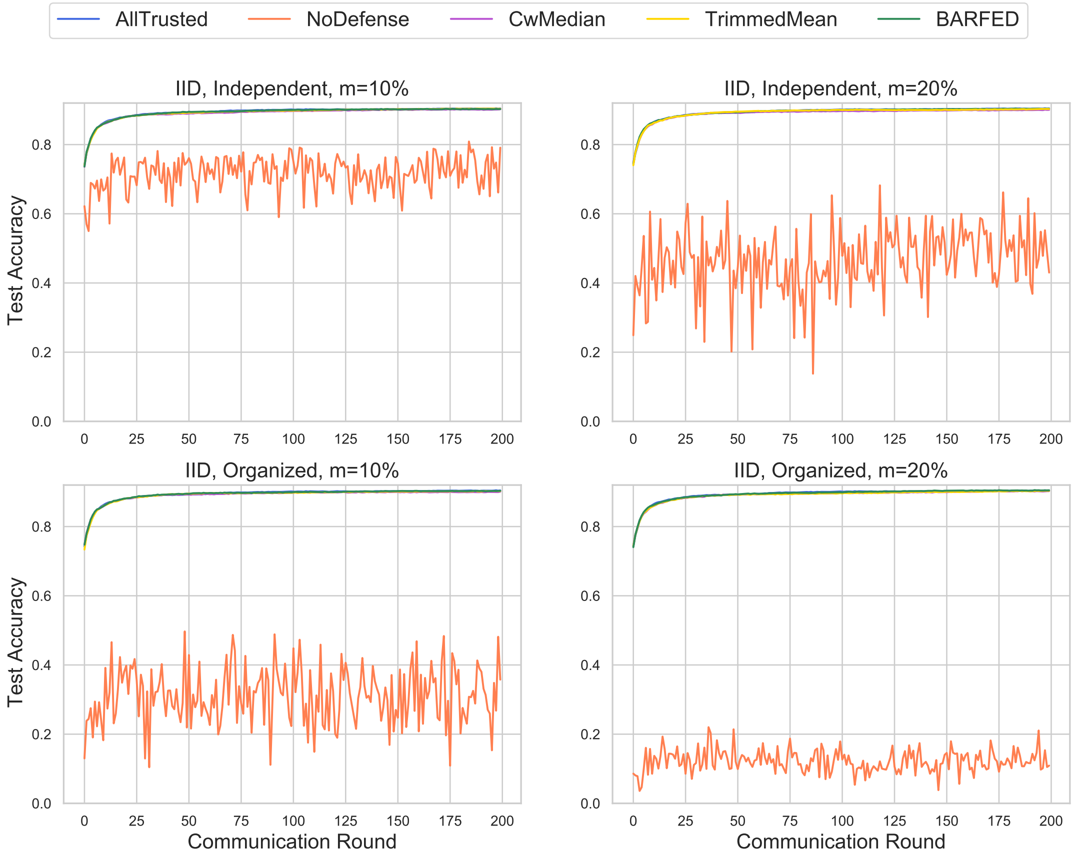

Figure 21: Accuracy curves for Fashion-MNIST under Byzantine attacks at different attacker ratios in the IID case.

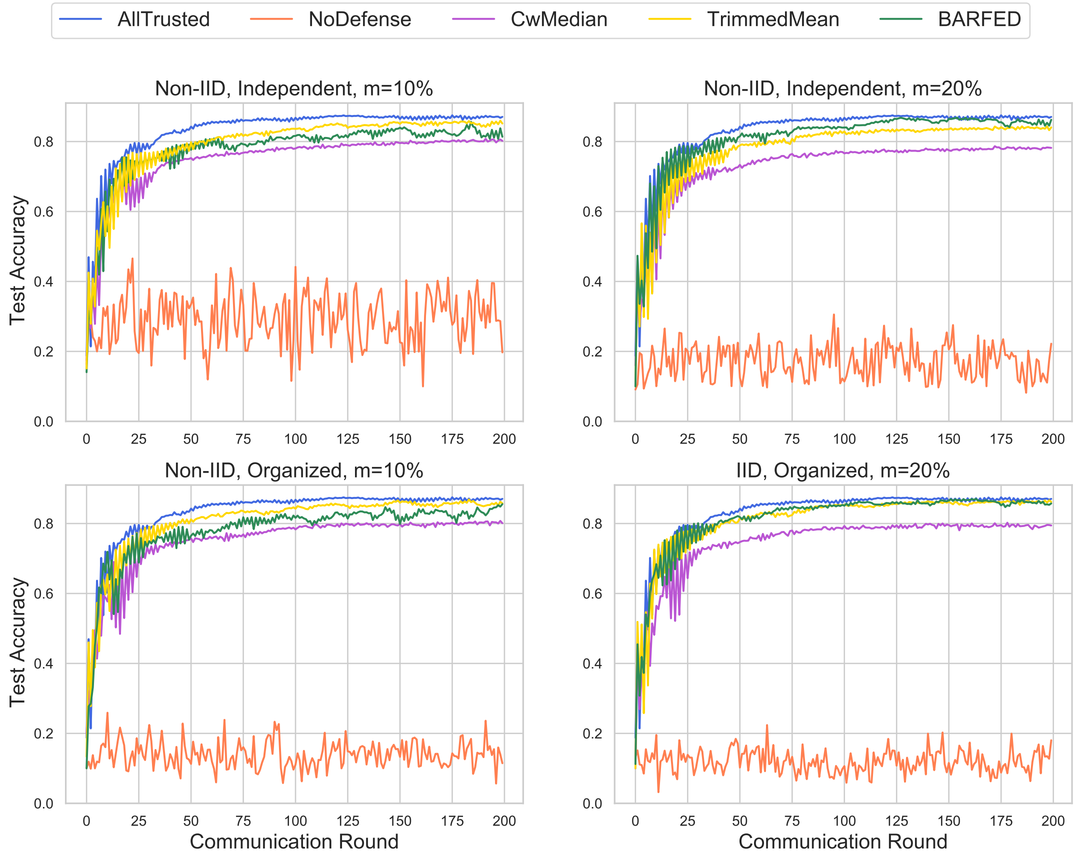

Figure 22: Accuracy curves for Fashion-MNIST under Byzantine attacks at different attacker ratios in the non-IID case.

6.4 CIFAR10

6.4.1 Label Flipping Attacks

Table 22 and Figure 23 present the results of the experiments carried out for label flipping attack on CIFAR10 when the collaborators' data are IID.

Table 23 and Figure 24 present the results of the experiments carried out for label flipping attack on CIFAR10 when the collaborators' data are non-IID.

Recall that, for the CIFAR10 data set, when all collaborators are trusted, the accuracy in the IID setting is 79.0% while it is 77.9% in the non-IID setting (see Table 8, Table 9).

The below comments are for the worst-case scenario where collaborators are organized and $m = 20\%$, unless otherwise stated.

For IID experiments, under label flipping attacks, accuracy degradation reaches up to 65.8%. CwMedian and TrimmedMean are able to recover some negative affect of the attack, and they can increase the accuracy to 73.4%. BARFED achieves a better accuracy of 77.2%.

For Non-IID experiments under label flipping attacks, the accuracies oscillate between 70.8% and 70.9%. CwMedian worsens the accuracy degradation. The accuracy score even decreases down to 54.5% (oscillates between 54.5% and 54.9%). TrimmedMean can increase the accuracy score up to 72.7%. BARFED outperforms the others in the non-IID case and increases the accuracies up to 76.9%.

Table 22: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in IID setting for CIFAR10. The best results are bold.

	Organized				**Independent**			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	72.7	72.7	65.8	65.9	72.8	72.8	69.7	69.8
CwMedian	75.6	75.7	73.3	73.4	73.8	73.8	73.5	73.6
TrimmedMean	75.6	75.7	73.3	73.4	73.8	73.8	73.5	73.6
BARFED	**76.2**	**76.2**	**77.0**	**77.2**	**77.7**	**77.8**	**75.6**	**75.7**

Table 23: Accuracy scores obtained under label flipping attacks with different malicious collaborator ratios in non-IID setting for CIFAR10. The best results are bold.

	Organized				Independent			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	75.2	75.3	70.8	70.9	74.4	74.4	70.2	70.3
CwMedian	55.0	55.9	54.5	54.9	56.7	57.2	52.0	52.7
TrimmedMean	76.0	76.0	72.6	72.7	75.4	75.4	74.0	74.1
BARFED	**78.0**	**78.1**	**76.8**	**76.9**	**77.3**	**77.4**	**76.2**	**76.3**

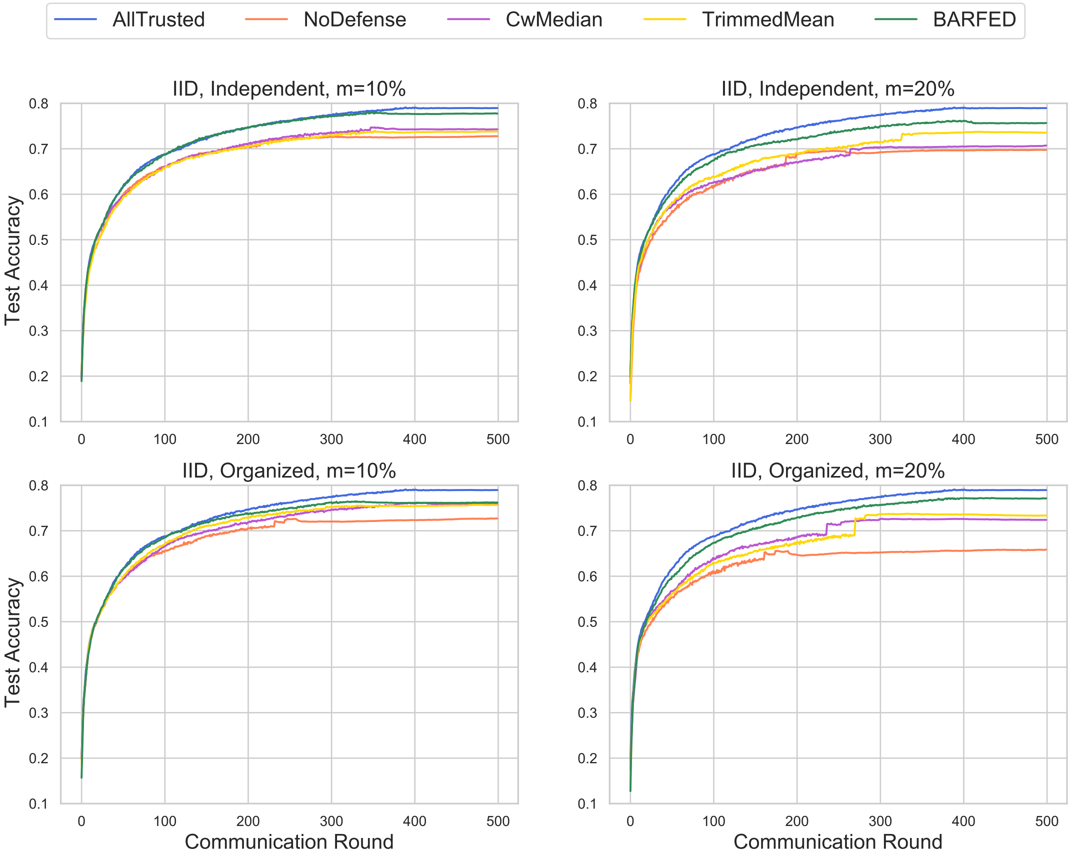

Figure 23: Accuracy curves for CIFAR10 under label flipping attacks at different attacker ratios in the IID case.

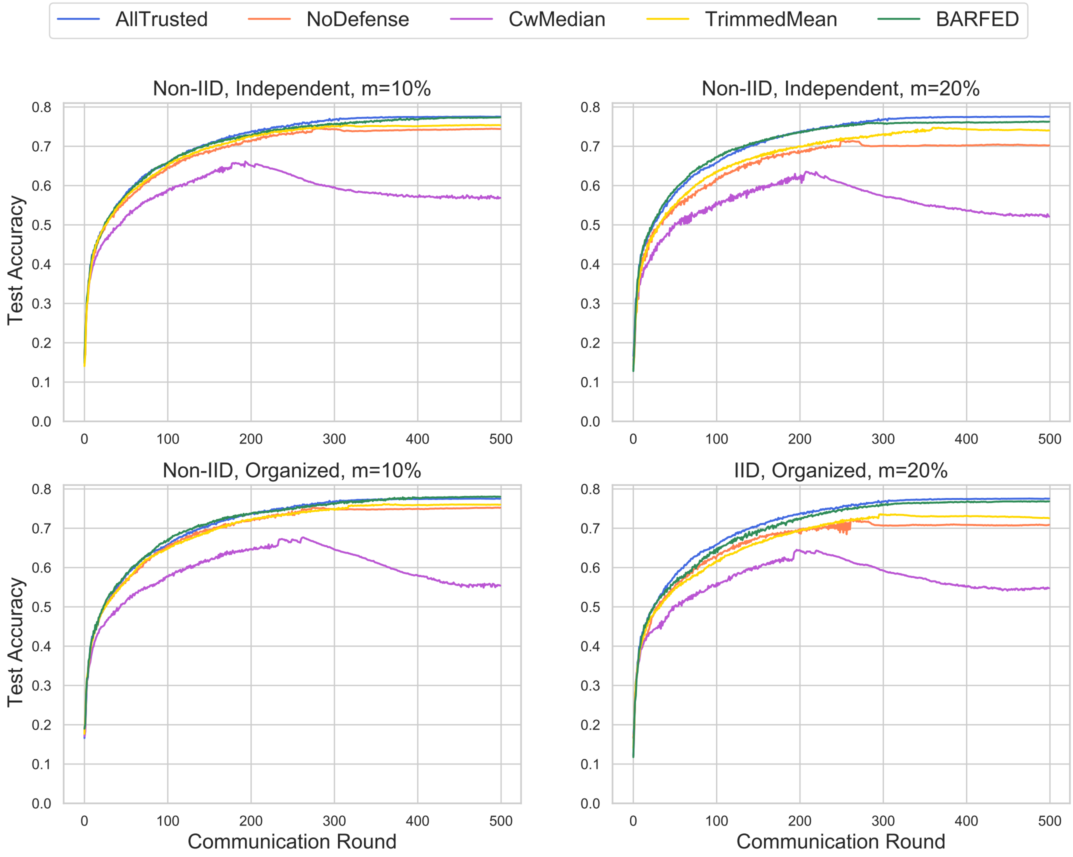

Figure 24: Accuracy curves for CIFAR10 under label flipping attacks at different attacker ratios in the non-IID case.

6.4.2 Byzantine Attacks

Table 24 and Figure 25 show the results of the Byzantine attack scenarios for CIFAR10 when the collaborators' data are IID.

Table 25 and Figure 26 show the results of the Byzantine attack scenarios for Fashion-MNIST when the collaborators' data are non-IID.

Regardless of the data distribution, the malicious collaborator ratio, and whether the attackers are organized, the Byzantine attacks cause huge performance losses for CIFAR10 experiments. The accuracy score drops down to 8.3%. For the IID cases, all methods can manage the reverse the attack and increase the accuracy scores. However, it cannot be said that one method outperforms the others. Nevertheless, BARFED achieves the highest scores when $m = 20\%$.

Table 24: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in IID setting for CIFAR10. The best results are bold.

	Organized				Independent			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	8.7	12.2	8.3	11.3	8.9	13.2	8.9	11.1
CwMedian	75.7	75.8	75.0	75.0	**77.6**	**77.7**	75.1	75.1
TrimmedMean	**78.8**	**78.9**	75.6	75.7	76.8	76.9	77.0	77.1
BARFED	77.8	77.9	**77.4**	**77.5**	77.1	77.2	**77.3**	**77.3**

For the non-IID cases, the accuracy drop reaches 7.9%. Although CwMedian recovers the negative effects of the attack up to a point, its success is limited. Contrary to other experiments, an unexpected situation developed for the non-IID cases of the CIFAR10 dataset. For other datasets, the scores of CwMedian when attackers are organized and $m = 10\%$ is better than when $m = 20\%$. However, the situation is the opposite of CIFAR10 experiments. CwMedian can increase the score to 72.0% when $m = 20\%$ while it can achieve only 59.0% accuracy when $m = 10\%$. CwMedian performs worse than TrimmedMean and BARFED significantly. BARFED provides the highest scores most of the time; however, TrimmedMean also can improve the accuracies as much as BARFED.

Table 25: Accuracy scores obtained under Byzantine attacks with different malicious collaborator ratios in non-IID setting for CIFAR10. The best results are bold.

	Organized				Independent			
	m=10%		m=20%		m=10%		m=20%	
	min	max	min	max	min	max	min	max
NoDefense	8.9	11.4	8.2	11.5	7.9	10.3	8.7	13.4
CwMedian	57.7	59.0	71.8	72.0	62.0	62.3	58.0	61.2
TrimmedMean	76.4	76.5	76.3	76.4	**76.6**	**76.6**	75.5	75.6
BARFED	**77.6**	**77.6**	**77.5**	**77.6**	75.2	75.3	**77.2**	**77.2**

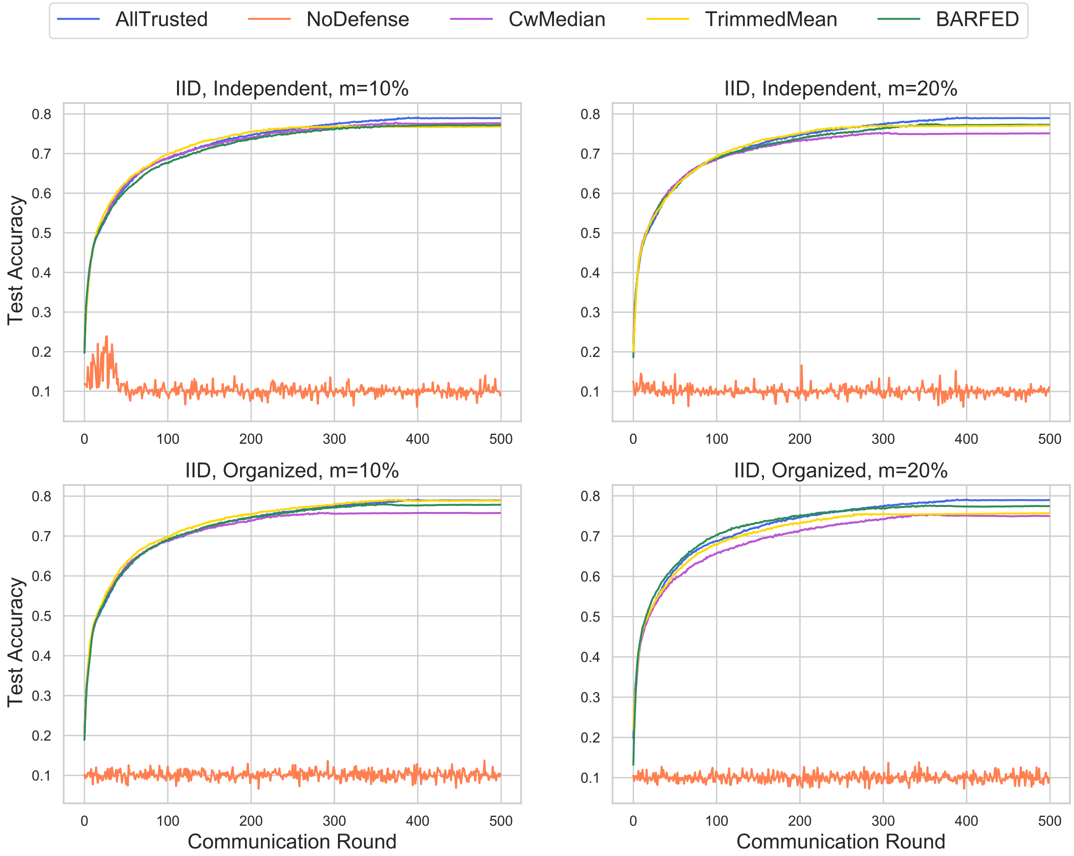

Figure 25: Accuracy curves for CIFAR10 under Byzantine attacks at different attacker ratios in the IID case.

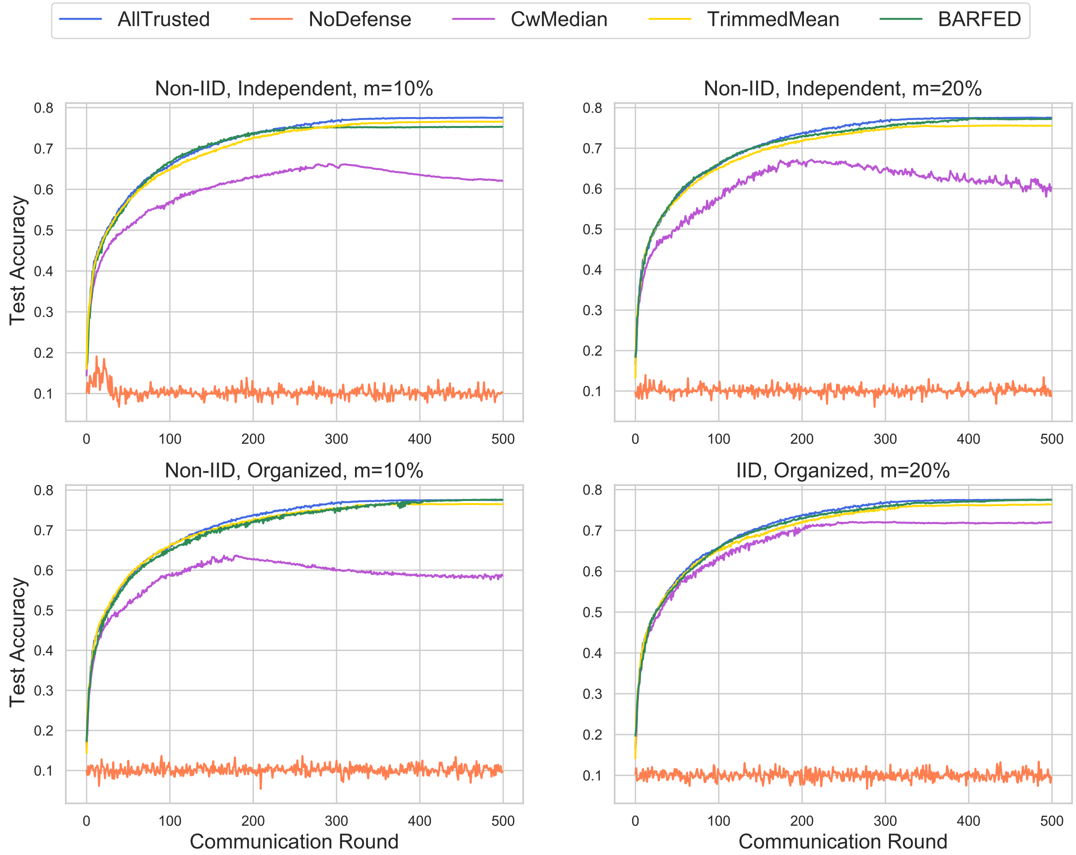

Figure 26: Accuracy curves for CIFAR10 under Byzantine attacks at different attacker ratios for the non-IID case.

6.4.3 Further Analysis

BARFED decides which collaborators are eligible to participate in the aggregation step by focusing on staying within the safe range for each layer of the model architecture. If a collaborator stays within the safe range calculated for all layers of the model architecture, it is marked as reliable by BARFED. If a collaborator falls outside this range at any layer, it will be excluded from the aggregation step.

The working mechanism of BARFED may raise concerns that too many collaborators may be discarded, and too much valuable knowledge related to these collaborators may be lost through the layers. However, if a collaborator is trusted, it generally tends to stay within the safe range in all their layers, while if a collaborator is malicious, it probably falls outside of this range in all or almost all layers.

Recall the distances of a randomly selected trusted participant to the global model are shown in the upper row of Figure 4 in Chapter 4. Here, the distance of the trusted participant to the main model remained within the safe range in all rounds. On the other hand, the lower row of the same figure shows that the distances of a malicious collaborator to the global model are out of the safe range for all layers and rounds.

Moreover, suppose there is "AND" operation effect on losing too many collaborators. In that case, it is expected to be observed on CIFAR10 experiments more obviously because CIFAR10 model architecture 5 has more layers than the models that are used for other datasets. However, the experiments show that the number of discarded collaborators is in line with the malicious collaborator ratio in the experiments.

Figure 27 illustrates the number of collaborators marked as reliable and included aggregation step versus the number of collaborators marked as outliers and discarded from aggregation in label flipping attacks for the CIFAR10 dataset with different scenarios. The number of discarded collaborators is in line with the malicious collaborator ratio.

Figure 28 illustrates the number of collaborators marked as reliable and included aggregation step versus the number of collaborators marked as outliers and discarded from aggregation in Byzantine attacks for the CIFAR10 dataset with different scenarios. The number of discarded collaborators is in line with the malicious collaborator ratio.

Figure 29 illustrates the average confusion matrix in CIFAR10 label flipping attacks throughout the communication rounds. Recall that, in the experiments there are 100 collaborators. Accordingly, BARFED is able to detect all collaborators in all scenarios except when attackers are organized and $m = 20\%$ in the non-IID setting.

Figure 30 illustrates the average confusion matrix in CIFAR10 Byzantine attacks. In Byzantine attacks, BARFED is able to detect all malicious collaborators and exclude them from the aggregation. In other words, no malicious collaborator is able to bypass BARFED. The number of trusted collaborators is excluded from the aggregation is reasonable in the IID setting and when attackers are independent in the non-IID setting. On the other hand, the number of trusted collaborators is excluded from

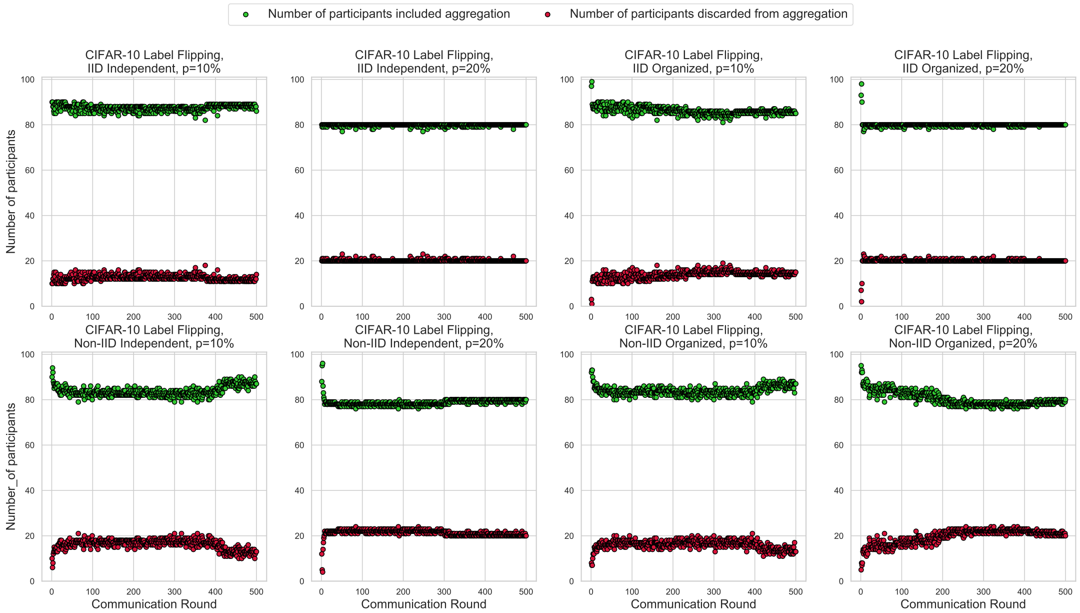

Figure 27: Number of collaborators marked as reliable and outlier in CIFAR10 label flipping attacks.

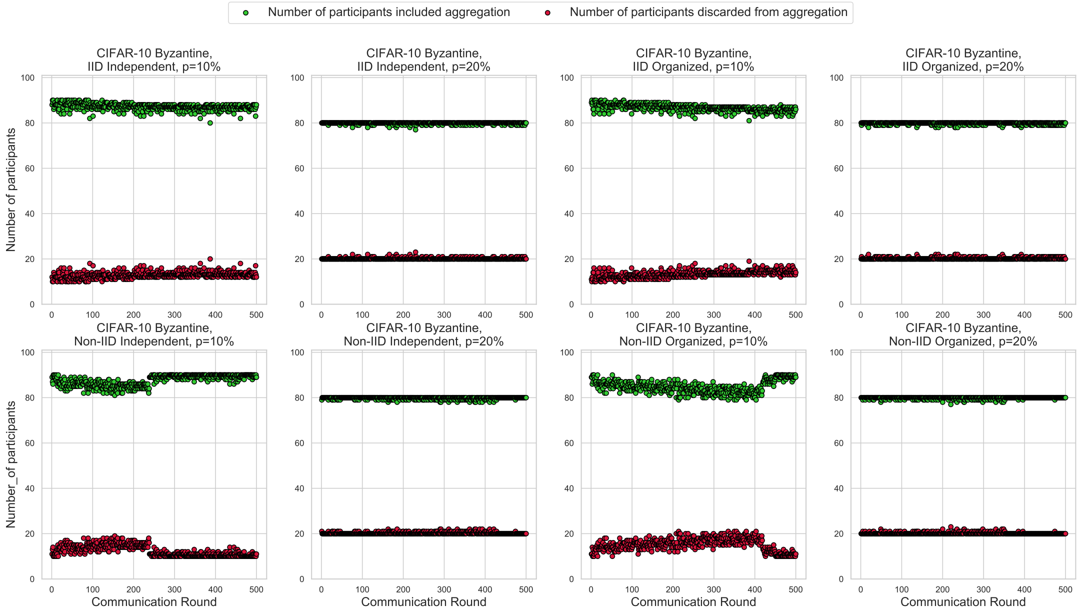

Figure 28: Number of collaborators marked as reliable and outlier in CIFAR10 Byzantine attacks.

the aggregation increases up to approximately 5 when attackers are organized in the non-IID setting. Still, considering the negative effects of malicious collaborators on the model performance, losing information about these collaborators is acceptable.

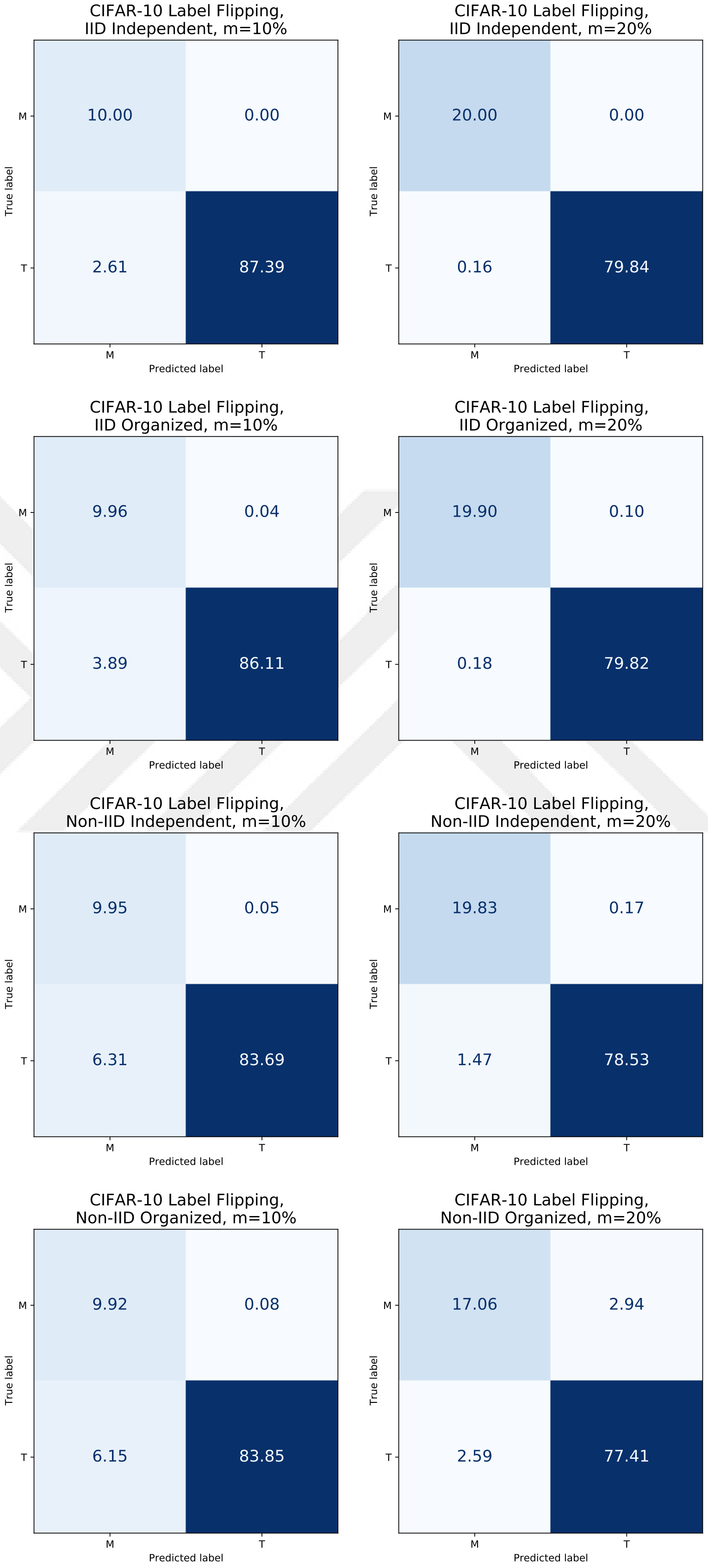

Figure 29: Average confusion matrix in CIFAR10 label flipping attacks.

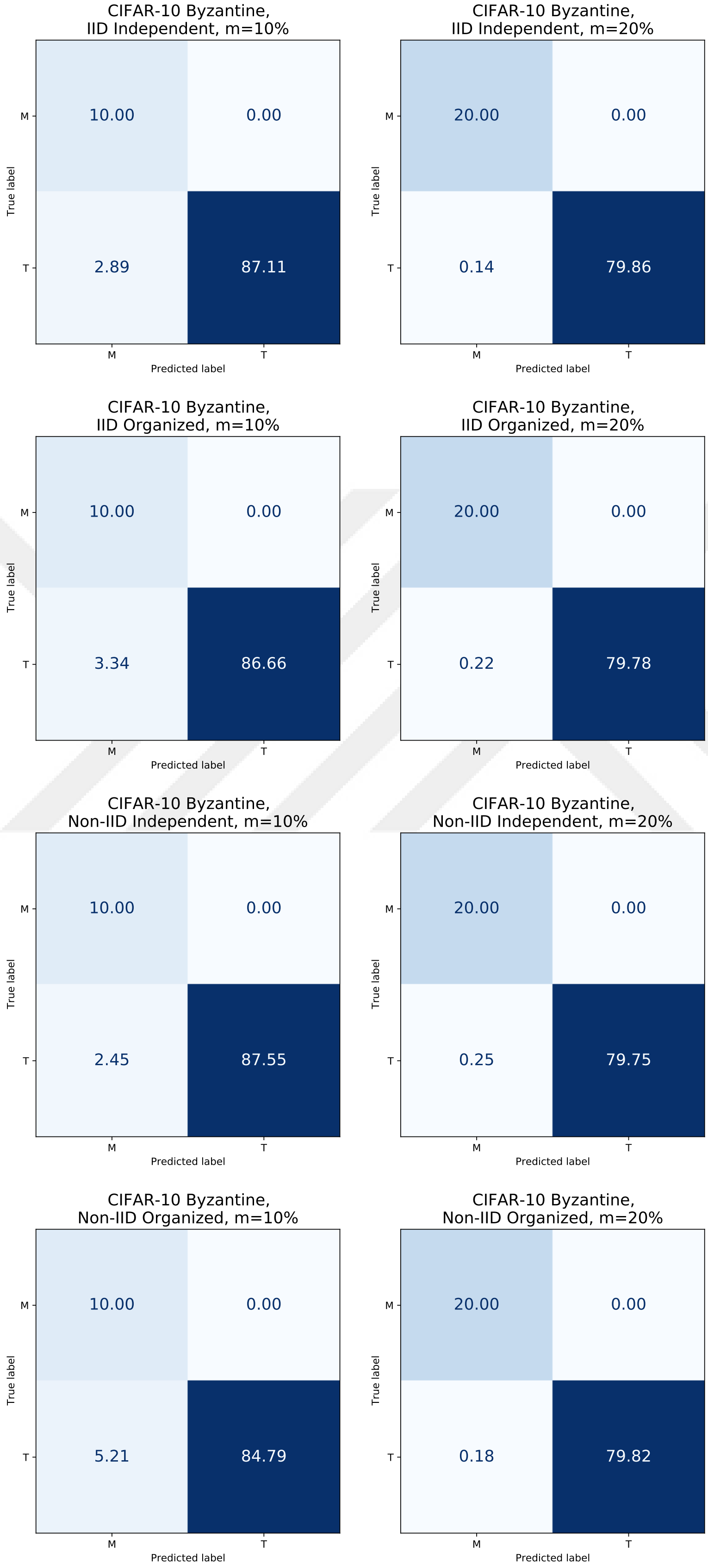

Figure 30: Average confusion matrix in CIFAR10 label flipping attacks.

6.5 Discussion

To sum up, non-IID attacks are more severe than IID attacks. For both IID and Non-IID cases, as the ratio of the malicious collaborator increases, the accuracy score of the FedAvg (NoDefense) degrades more. When malicious collaborators are organized, the degradation in performance becomes more severe.

Some recent studies ([16, 17, 12]) show that Byzantine attacks negatively affect model performances more than data poisoning attacks. The experiments showed that the model performances decreased dramatically under the Byzantine attack, which is in line with previous studies.

Most of the time, BARFED gets marginally better scores than CwMedian and TrimmedMean in the IID experiments of the MNIST-2NN, MNIST-CNN, and Fashion-MNIST for label flipping attacks. Still, the difference can be considered insignificant. However, BARFED achieves distinctively better accuracies in the IID cases of CIFAR10.

On the other hand, when the collaborators' data are non-IID, the accuracies that are achieved by CwMedian are worse than BARFED and even FedAvg (no defense case) in MNIST-2NN and Fashion MNIST for label flipping attack experiments. BARFED is able to defend against the harmful effects of the attack successfully and achieves accuracy scores close to when all collaborators are trusted cases (no attack case). BARFED is generally better than TrimmedMean in label flipping attacks, but they get similar scores in Byzantine attacks.

However, it is worth remembering that TrimmedMean requires information of the malicious collaborator ratio in the system while BARFED does not make such an assumption.

CHAPTER 7

CONCLUSIONS

In this thesis, I propose an assumption-free defense strategy against Byzantine attacks. I have evaluated the performance of my method against Byzantine attacks as well as other attack types in various scenarios. I have also compared my method with baseline methods.

The primary contributions of this thesis are listed below.

- I propose a method called BARFED that is resistant to Byzantine attacks. BARFED does not make any assumptions inconsistent with the characteristics of federated learning. In particular, BARFED does not need to know the malicious collaborator ratio in the system. Moreover, BARFED can eliminate the harmful effects of the attacks in both IID and Non-IID cases. BARFED is independent of the gradient update similarity of the collaborators.
- For the performance evaluation of BARFED, extensive experiments have been carried out. These experiments cover many scenarios. The effects of the distribution of data, whether the attackers are organized, and different types of attacks have been investigated on different datasets with different model architectures.
- I show that Byzantine attacks are generally more severe than label flipping attacks. The performance degradation caused by such attacks gets worse when the collaborators' data are non-IID, which is common in most of the FL settings.
- I show that BARFED can successfully eliminate harmful effects of attacks and stabilizes the convergence of attacks both for IID and Non-IID data distribution while the CwMedian cannot handle the attacks in the Non-IID setting. CwMedian could resist the attack in only IID cases. For non-IID cases, it could show only a slight improvement or worsen the performance degradation.

Despite all the contributions listed above, there are aspects of this study that need further research and development.

BARFED is mainly based on the elimination of outliers. I used box plot outlier elimination technique but there are some other techniques such as z-score [89], isolation forests [90], local outlier factor (LOF) [91] or DBSCAN [92] for outlier detection.

The integration and performance analysis of these outlier detection methods should be examined.

In this thesis, IID and non-IID discrimination is made based on whether the label distributions in collaborators are balanced or imbalanced. For example, for the MNIST dataset, the collaborators have only two classes in their local datasets. However, different approaches that are based on features can also be applied for non-IID and IID definitions. For example, a distinction can be made for MNIST on the basis of writing style or the person who wrote the number. This type of FL setting definition needs further work.

In the current version of BARFED, a collaborator is included in the aggregation step of the FL round if the collaborator is in the safe range at all layers. If the collaborator is labeled as an outlier in any layer, I exclude it from the calculation step. A few collaborators may be marked as outliers for each layer, but due to the "and" operation that I use in the algorithm, too many collaborators may be excluded in the aggregation step. The experiments have shown that if a collaborator is malicious, it is usually an outlier at all or almost all layers. However, with more complex model architectures including a large number of layers, this may be an important issue to deal with. Therefore, the approach should be reconsidered and redesigned for the more complex model architectures such as VGG[93], ResNet [94], DenseNet[95], MobileNet[96] and EfficientNet[97].

Another issue that needs to be examined is the effect of the number of collaborators in the system. Since collaborator elimination is performed on specific statistics such as Q1 and Q3, if there are few participants in the system, these values may vary, which may adversely affect the performance of the method in distinguishing malicious collaborators. However, as the number of collaborators increases, I expect performance consistency to increase as the values obtained for these statistics will be more unbiased.

Moreover, adversarial attacks can be a problem in federated settings. Adversarial attacks can be seen as a mix of label flipping attacks and adding noise to the training data. Since they aim to change predicted class by manipulating training data for example adding noise to the pixels of the image. I think that BARFED can recover these types of attacks in the federated setting but it needs further investigation.

Since BARFED is based on the elimination of outliers, it can only resist attacks involving a certain percentage of malicious collaborators. If there are too many malicious collaborators in the system, they might no longer be perceived as outliers. Therefore, BARFED will not be able to eliminate them in the aggregation step. Similarly, in other outlier-based methods, increasing the ratio of malicious collaborators that can be tolerated is an issue that needs to be developed for BARFED too.

REFERENCES

[1] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, “Federated learning with personalization layers,” *arXiv preprint arXiv:1912.00818*, 2019.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[3] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, “Advances and open problems in federated learning,” *arXiv preprint arXiv:1912.04977*, 2019.

[4] M. Duan, D. Liu, X. Chen, Y. Tan, J. Ren, L. Qiao, and L. Liang, “Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications,” in *2019 IEEE 37th International Conference on Computer Design (ICCD)*, pp. 246–254, IEEE, 2019.

[5] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, “Federated optimization: Distributed machine learning for on-device intelligence,” *arXiv preprint arXiv:1610.02527*, 2016.

[6] R. Pathak and M. J. Wainwright, “Fedsplit: an algorithmic framework for fast federated optimization,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 7057–7066, Curran Associates, Inc., 2020.

[7] H. Yuan and T. Ma, “Federated accelerated stochastic gradient descent,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[8] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 7611–7623, Curran Associates, Inc., 2020.

[9] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, “Adaptive federated optimization,” in *International Conference on Learning Representations*, 2021.

[10] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Analyzing federated learning through an adversarial lens,” in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.

[11] C. Fung, C. J. Yoon, and I. Beschastnikh, “Mitigating sybils in federated learning poisoning,” *arXiv preprint arXiv:1808.04866*, 2018.

[12] F. Sattler, K.-R. Müller, T. Wiegand, and W. Samek, "On the byzantine robustness of clustered federated learning," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8861–8865, IEEE, 2020.

[13] S. Shen, S. Tople, and P. Saxena, "Auror: Defending against poisoning attacks in collaborative deep learning systems," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pp. 508–519, 2016.

[14] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 118–128, 2017.

[15] 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*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 3521–3530, PMLR, 10–15 Jul 2018.

[16] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1605–1622, 2020.

[17] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, pp. 2938–2948, PMLR, 2020.

[18] H. Wang, K. Sreenivasan, S. Rajput, H. Vishwakarma, S. Agarwal, J.-y. Sohn, K. Lee, and D. Papailiopoulos, "Attack of the tails: Yes, you really can backdoor federated learning," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, pp. 16070–16084, Curran Associates, Inc., 2020.

[19] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, "Draco: Byzantine-resilient distributed training via redundant gradients," in *International Conference on Machine Learning*, pp. 903–912, PMLR, 2018.

[20] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *International Conference on Machine Learning*, pp. 6893–6901, PMLR, 2019.

[21] C. Xie, S. Koyejo, and I. Gupta, "Zeno++: Robust fully asynchronous sgd," in *International Conference on Machine Learning*, pp. 10495–10503, PMLR, 2020.

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

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

Systems (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[24] S. Rajput, H. Wang, Z. Charles, and D. Papailiopoulos, “Detox: A redundancy-based framework for faster and more robust gradient aggregation,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[25] B. Custers, A. M. Sears, F. Dechesne, I. Georgieva, T. Tani, and S. Van der Hof, *EU personal data protection in policy and practice*, vol. 29. Springer, 2019.

[26] P. Voigt and A. Von dem Bussche, “The eu general data protection regulation (gdpr),” *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, p. 3152676, 2017.

[27] W. House, “Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy,” *White House, Washington, DC*, pp. 1–62, 2012.

[28] B. M. Gaff, H. E. Sussman, and J. Geetter, “Privacy and big data,” *Computer*, vol. 47, no. 6, pp. 7–9, 2014.

[29] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, “Federated learning in mobile edge networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[30] P. Smyth, M. Welling, and A. Asuncion, “Asynchronous distributed learning of topic models,” in *Advances in Neural Information Processing Systems* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), vol. 21, Curran Associates, Inc., 2009.

[31] S. Niknam, H. S. Dhillon, and J. H. Reed, “Federated learning for wireless communications: Motivation, opportunities, and challenges,” *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[32] Q. Yang, Y. Liu, T. Chen, and Y. Tong, “Federated machine learning: Concept and applications,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[33] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, “Federated learning of predictive models from federated electronic health records,” *International journal of medical informatics*, vol. 112, pp. 59–67, 2018.

[34] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[35] Q. Dou, T. Y. So, M. Jiang, Q. Liu, V. Vardhanabhuti, G. Kaissis, Z. Li, W. Si, H. H. Lee, K. Yu, *et al.*, "Federated deep learning for detecting covid-19 lung abnormalities in ct: a privacy-preserving multinational validation study," *NPJ digital medicine*, vol. 4, no. 1, pp. 1–11, 2021.

[36] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, *et al.*, "Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[37] N. Rieke, J. Hancox, W. Li, F. Milletari, H. R. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. A. Landman, K. Maier-Hein, *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[38] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "Loadaboost: Loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data," *Plos one*, vol. 15, no. 4, p. e0230706, 2020.

[39] W. Li, F. Milletarì, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso, *et al.*, "Privacy-preserving federated brain tumour segmentation," in *International workshop on machine learning in medical imaging*, pp. 133–141, Springer, 2019.

[40] X. Li, Y. Gu, N. Dvornek, L. H. Staib, P. Ventola, and J. S. Duncan, "Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results," *Medical Image Analysis*, vol. 65, p. 101765, 2020.

[41] J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated learning for healthcare informatics," *Journal of Healthcare Informatics Research*, vol. 5, no. 1, pp. 1–19, 2021.

[42] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 83–93, 2020.

[43] T. Li, L. Song, and C. Fragouli, "Federated recommendation system via differential privacy," in *2020 IEEE International Symposium on Information Theory (ISIT)*, pp. 2592–2597, IEEE, 2020.

[44] Y. Lin, P. Ren, Z. Chen, Z. Ren, D. Yu, J. Ma, M. d. Rijke, and X. Cheng, "Meta matrix factorization for federated rating predictions," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 981–990, 2020.

[45] L. Yang, B. Tan, V. W. Zheng, K. Chen, and Q. Yang, "Federated recommendation systems," in *Federated Learning*, pp. 225–239, Springer, 2020.

[46] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: opportunities and solutions," *IEEE Network*, vol. 35, no. 2, pp. 152–159, 2021.

[47] W. Y. B. Lim, J. Huang, Z. Xiong, J. Kang, D. Niyato, X.-S. Hua, C. Leung, and C. Miao, “Towards federated learning in uav-enabled internet of vehicles: A multi-dimensional contract-matching approach,” *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[48] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. R. Cui, “Fedloc: Federated learning framework for data-driven cooperative localization and location data processing,” *IEEE Open Journal of Signal Processing*, vol. 1, pp. 187–215, 2020.

[49] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, “Energy demand prediction with federated learning for electric vehicle networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.

[50] D. Ye, R. Yu, M. Pan, and Z. Han, “Federated learning in vehicular edge computing: A selective model aggregation approach,” *IEEE Access*, vol. 8, pp. 23920–23935, 2020.

[51] G. Long, Y. Tan, J. Jiang, and C. Zhang, “Federated learning for open banking,” in *Federated learning*, pp. 240–254, Springer, 2020.

[52] Y. Liu, Z. Ai, S. Sun, S. Zhang, Z. Liu, and H. Yu, “Fedcoin: A peer-to-peer payment system for federated learning,” in *Federated Learning*, pp. 125–138, Springer, 2020.

[53] D. Guliani, F. Beaufays, and G. Motta, “Training speech recognition models with federated learning: A quality/cost framework,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3080–3084, IEEE, 2021.

[54] X. Cui, S. Lu, and B. Kingsbury, “Federated acoustic modeling for automatic speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6748–6752, IEEE, 2021.

[55] C. Tan, D. Jiang, H. Mo, J. Peng, Y. Tong, W. Zhao, C. Chen, R. Lian, Y. Song, and Q. Xu, “Federated acoustic model optimization for automatic speech recognition,” in *International Conference on Database Systems for Advanced Applications*, pp. 771–774, Springer, 2020.

[56] D. Jiang, C. Tan, J. Peng, C. Chen, X. Wu, W. Zhao, Y. Song, Y. Tong, C. Liu, Q. Xu, *et al.*, “A gdpr-compliant ecosystem for speech recognition with transfer, federated, and evolutionary learning,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 3, pp. 1–19, 2021.

[57] T.-M. H. Hsu, H. Qi, and M. Brown, “Federated visual classification with real-world data distribution,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pp. 76–92, Springer, 2020.

[58] Y. Liu, A. Huang, Y. Luo, H. Huang, Y. Liu, Y. Chen, L. Feng, T. Chen, H. Yu, and Q. Yang, "Fedvision: An online visual object detection platform powered by federated learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13172–13179, 2020.

[59] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated learning for keyword spotting," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345, IEEE, 2019.

[60] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2019.

[61] D. Liu, D. Dligach, and T. Miller, "Two-stage federated phenotyping and patient representation learning," in *Proceedings of the conference. Association for Computational Linguistics. Meeting*, vol. 2019, p. 283, NIH Public Access, 2019.

[62] 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.

[63] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations*, 2020.

[64] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in neural information processing systems*, vol. 26, pp. 315–323, 2013.

[65] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020.

[66] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics* (S. Chiappa and R. Calandra, eds.), vol. 108 of *Proceedings of Machine Learning Research*, pp. 2021–2031, PMLR, 26–28 Aug 2020.

[67] Z. Tao and Q. Li, "esgd: Communication efficient distributed deep learning on the edge," in {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[68] N. Strom, "Scalable distributed dnn training using commodity gpu cloud computing," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[69] "Tensorflow federated." [Online]. Available: `https://www.tensorflow.org/federated?hl=en/`. [Accessed: Aug. 11, 2021].

[70] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[71] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[72] OpenMined, "Pysyft." [Online]. Available: `https://github.com/OpenMined/PySyft`. [Accessed: Aug. 11, 2021].

[73] NVIDIA, "Clara training framework." [Online]. Available: `https://docs.nvidia.com/clara/clara-train-sdk/pt/index.html`. [Accessed: Aug. 11, 2021].

[74] G. A. Reina, A. Gruzdev, P. Foley, O. Perepelkina, M. Sharma, I. Davidyuk, I. Trushkin, M. Radionov, A. Mokrov, D. Agapov, *et al.*, "Openfl: An open-source framework for federated learning," *arXiv preprint arXiv:2105.06413*, 2021.

[75] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečnỳ, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.

[76] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, P. P. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," *arXiv preprint arXiv:2007.14390*, 2020.

[77] WeBank, "Federated ai technology enabler." [Online]. Available: `https://fate.fedai.org`. [Accessed: Aug. 11, 2021].

[78] Y. Ma, D. Yu, T. Wu, and H. Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019.

[79] PaddlePaddle, "Paddlefl." [Online]. Available: `https://github.com/PaddlePaddle/PaddleFL`. [Accessed: Aug. 11, 2021].

[80] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[81] L. Lyu, H. Yu, and Q. Yang, “Threats to federated learning: A survey,” *CoRR*, vol. abs/2003.02133, 2020.

[82] L. Zhu, Z. Liu, and S. Han, “Deep leakage from gradients,” in *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.

[83] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, “Data poisoning attacks against federated learning systems,” in *European Symposium on Research in Computer Security*, pp. 480–501, Springer, 2020.

[84] X. Cao, M. Fang, J. Liu, and N. Z. Gong, “Fltrust: Byzantine-robust federated learning via trust bootstrapping,” *arXiv preprint arXiv:2012.13995*, 2020.

[85] J. W. Tukey *et al.*, *Exploratory data analysis*, vol. 2. Reading, Mass., 1977.

[86] Y. LeCun, “The mnist database of handwritten digits,” *http://yann. lecun. com/exdb/mnist/*, 1998.

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

[88] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[89] V. Barnett and T. Lewis, “Outliers in statistical data,” *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, 1984.

[90] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 eighth ieee international conference on data mining*, pp. 413–422, IEEE, 2008.

[91] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

[92] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *kdd*, vol. 96, pp. 226–231, 1996.

[93] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[94] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[95] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[96] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[97] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

[98] M. Friedman, “The use of ranks to avoid the assumption of normality implicit in the analysis of variance,” *Journal of the american statistical association*, vol. 32, no. 200, pp. 675–701, 1937.

[99] “Friedman test.” [Online]. Available: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.friedmanchisquare.html`. [Accessed: Sept. 29, 2021].

[100] P. B. Nemenyi, *Distribution-free multiple comparisons.* Princeton University, 1963.

[101] “posthoc nemenyi friedman.” [Online]. Available: `https://scikit-posthocs.readthedocs.io/en/latest/generated/scikit_posthocs.posthoc_nemenyi_friedman/`. [Accessed: Sept. 29, 2021].

APPENDIX A

PERFORMANCE COMPARISONS OF BARFED WITH OTHER BASELINE METHODS

This chapter presents the performance comparison of BARFED, TrimmedMean, and CwMedian under different attack scenarios. For the comparison, hypothesis testing is performed. The obtained accuracies are recorded for each data set, each defense mechanism, and each attack scenario. Since the recorded data is paired, the Friedman Chi-Square Test [98, 99] is used to compare all methods. If there is a statistically significant difference between these methods, the Nemenyi Friedman Test [100, 101] is performed for posthoc analysis and pairwise comparison. $\alpha = 0.05$

A.1 MNIST 2NN

A.1.1 Label Flipping Attacks

A.1.1.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [97.4, 97.4, 97.2, 97.3, 97.4, 97.5, 97.0, 97.1]
TrimmedMean = [97.5, 97.5, 96.9, 97.0, 97.5, 97.5, 97.1, 97.1]
BARFED = [97.6, 97.7, 97.4, 97.5, 97.6, 97.6, 97.4, 97.5]

$\bar{x}_{CwMedian} = 97.29$, $\bar{x}_{TrimmedMean} = 97.26$, $\bar{x}_{BARFED} = 97.54$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0015$ < 0.05).

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and CwMedian ($p_{val} = 0.8575 > 0.05$) but BARFED out-

Table 26: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in IID setting of MNIST-2NN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.8575	0.0033
TrimmedMean	0.8575	1.0000	0.0164
BARFED	0.0033	0.0164	1.0000

performs CwMedian ($p_{one-tailed-val} = (0.0033/2) < 0.05$) and TrimmedMean ($p_{one-tailed-val} = (0.0164/2) < 0.05$).

A.1.1.2 Non-IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [75.1, 83.8, 67.7, 75.0, 80.8, 83.3, 67.8, 75.9]
TrimmedMean = [94.6, 95.5, 79.9, 87.7, 95.1, 95.4, 80.9, 89.7]
BARFED = [96.1, 96.3, 95.6, 96.1, 96.0, 96.3, 95.5, 96.1]

$\bar{x}_{CwMedian} = 76.17$, $\bar{x}_{TrimmedMean} = 89.85$, $\bar{x}_{BARFED} = 96.00$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
($\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED}$)
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

Table 27: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in non-IID setting of MNIST-2NN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.1123	0.0010
TrimmedMean	0.1123	1.0000	0.1123
BARFED	0.0010	0.1123	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.1123 > 0.05$) but BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0010/2) < 0.05$).

A.1.2 Byzantine Attacks

A.1.2.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [97.2, 97.2, 97.3, 97.4, 97.2, 97.3, 97.3, 97.3]
TrimmedMean = [97.4, 97.5, 97.4, 97.5, 97.5, 97.5, 97.4, 97.4]
BARFED = [97.5, 97.5, 97.5, 97.6, 97.5, 97.5, 97.5, 97.6]

$\bar{x}_{CwMedian} = 97.28$, $\bar{x}_{TrimmedMean} = 97.45$, $\bar{x}_{BARFED} = 97.52$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0006$ < 0.05).

Table 28: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in IID setting of MNIST-2NN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0462	0.0010
TrimmedMean	0.0462	1.0000	0.4256
BARFED	0.0010	0.4256	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.4256 > 0.05$) and they both outperform CwMedian ($p_{one-tailed-val} = (0.0462/2) < 0.05$, $p_{one-tailed-val} = (0.0010/2) <$ 0.05).

A.1.2.2 Non-IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [85.6, 89.5, 90.3, 92.7, 83.4, 88.8, 89.3, 90.9]
TrimmedMean = [95.8, 96.1, 95.4, 95.6, 95.9, 96.1, 94.8, 95.0]
BARFED = [96.1, 96.2, 95.9, 96.1, 96.1, 96.2, 95.9, 96.1]

$\bar{x}_{CwMedian} = 88.81$, $\bar{x}_{TrimmedMean} = 95.59$, $\bar{x}_{BARFED} = 96.08$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

Table 29: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in non-IID setting of MNIST-2NN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.1123	0.0010
TrimmedMean	0.1123	1.0000	0.1123
BARFED	0.0010	0.1123	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.1123 > 0.05$) but BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0010/2) < 0.05$).

A.2 MNIST CNN

A.2.1 Label Flipping Attacks

A.2.1.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [98.9, 98.9, 98.8, 98.8, 98.9, 98.9, 98.9, 98.9]
TrimmedMean = [98.9, 98.9, 98.8, 98.9, 99.0, 99.0, 98.9, 99.0]
BARFED = [98.9, 98.9, 98.9, 98.9, 98.9, 98.9, 98.9, 98.9]

$\bar{x}_{CwMedian} = 98.88$, $\bar{x}_{TrimmedMean} = 98.92$, $\bar{x}_{BARFED} = 98.90$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
($\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED}$)
H_1: At least one of them is different.

There is no statistically significant difference between CwMedian, TrimmedMean and BARFED according to the result of the the Friedman Chi-Square test ($p_{val} = 0.0907$ > 0.05).

A.2.1.2 Non-IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [96.4, 96.7, 91.4, 93.1, 95.4, 95.9, 93.5, 94.2]
TrimmedMean = [98.5, 98.6, 97.3, 97.6, 98.6, 98.6, 97.2, 97.6]
BARFED = [98.6, 98.7, 98.5, 98.6, 98.7, 98.8, 98.6, 98.7]

$\bar{x}_{CwMedian} = 94.58$, $\bar{x}_{TrimmedMean} = 98.00$, $\bar{x}_{BARFED} = 98.65$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

Table 30: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in non-IID setting of MNIST-CNN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.1123	0.0010
TrimmedMean	0.1123	1.0000	0.1123
BARFED	0.0010	0.1123	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.1123 > 0.05$) but BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0010/2) < 0.05$).

A.2.2 Byzantine Attacks

A.2.2.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [98.9, 98.9, 98.9, 98.9, 98.9, 98.9, 98.8, 98.8]
TrimmedMean = [99.0, 99.0, 98.9, 99.0, 98.9, 99.0, 98.9, 98.9]
BARFED = [98.9, 98.9, 98.8, 98.8, 98.9, 99.0, 98.8, 98.8]

$\bar{x}_{CwMedian} = 98.88$, $\bar{x}_{TrimmedMean} = 98.95$, $\bar{x}_{BARFED} = 98.86$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0071$ < 0.05).

The posthoc analysis shows that there is no statistically significant difference between CwMedian and BARFED ($p_{val} = 0.9000 > 0.05$) and TrimmedMean outperforms BARFED ($p_{one-tailed-val} = (0.0462/2) < 0.05$).

Table 31: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in IID setting of MNIST-CNN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0849	0.9000
TrimmedMean	0.0849	1.0000	0.0462
BARFED	0.9000	0.0462	1.0000

A.2.2.2 Non-IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [97.4, 97.5, 97.6, 97.7, 97.0, 97.2, 96.8, 97.1]
TrimmedMean = [98.8, 98.8, 98.6, 98.7, 98.8, 98.8, 98.5, 98.6]
BARFED = [98.7, 98.8, 98.6, 98.7, 98.7, 98.8, 98.6, 98.7]

$\bar{x}_{CwMedian} = 97.29$, $\bar{x}_{TrimmedMean} = 98.70$, $\bar{x}_{BARFED} = 98.70$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same. ($\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED}$)
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0011$ < 0.05).

Table 32: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in non-IID setting of MNIST-CNN architecture.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0076	0.0076
TrimmedMean	0.0076	1.0000	0.9000
BARFED	0.0076	0.9000	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.9000 > 0.05$) and BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0076/2) < 0.05$).

A.3 Fashion MNIST

A.3.1 Label Flipping Attacks

A.3.1.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [89.8, 89.9, 88.6, 88.7, 89.6, 89.7, 89.2, 89.3]
TrimmedMean = [90.0, 90.1, 88.8, 88.9, 89.9, 90.0, 89.0, 89.2]
BARFED = [90.5, 90.7, 90.3, 90.4, 90.2, 90.3, 90.2, 90.3]

$\bar{x}_{CwMedian} = 89.35$, $\bar{x}_{TrimmedMean} = 89.49$, $\bar{x}_{BARFED} = 90.36$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.00015$ < 0.05).

Table 33: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in IID setting of Fashion MNIST.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.5715	0.0014
TrimmedMean	0.5715	1.0000	0.0333
BARFED	0.0014	0.0333	1.0000

The posthoc analysis shows that there is no statistically significant difference between CwMedian and TrimmedMean ($p_{val} = 0.5715 > 0.05$) and BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0014/2) < 0.05$) and TrimmedMean ($p_{one-tailed-val} = (0.0333/2) < 0.05$).

A.3.1.2 Non-IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [79.9, 80.6, 76.0, 76.7, 78.7, 79.6, 77.4, 78.5]
TrimmedMean = [86.7, 87.5, 82.7, 83.4, 85.8, 86.7, 83.8, 84.3]
BARFED = [87.7, 88.8, 84.2, 87.6, 87.5, 88.5, 85.1, 87.6]

$\bar{x}_{CwMedian} = 78.43$, $\bar{x}_{TrimmedMean} = 85.11$, $\bar{x}_{BARFED} = 87.12$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

Table 34: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in non-IID setting of Fashion MNIST.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.1123	0.0010
TrimmedMean	0.1123	1.0000	0.1123
BARFED	0.0010	0.1123	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.1123 > 0.05$) and BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0010/2) < 0.05$).

A.3.2 Byzantine Attacks

A.3.2.1 IID Cases

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [89.9, 90.1, 90.1, 90.2, 90.0, 90.2, 90.0, 90.1]
TrimmedMean = [90.2, 90.3, 90.2, 90.3, 90.4, 90.4, 90.2, 90.3]
BARFED = [90.2, 90.3, 90.4, 90.5, 90.2, 90.3, 90.3, 90.4]

$\bar{x}_{CwMedian} = 90.07$, $\bar{x}_{TrimmedMean} = 90.29$, $\bar{x}_{BARFED} = 90.32$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0015$ < 0.05).

Table 35: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in IID setting of Fashion MNIST.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0164	0.0033
TrimmedMean	0.0164	1.0000	0.8575
BARFED	0.0033	0.8575	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.8575 > 0.05$) and they both outperform CwMedian ($p_{one_tailed_val} = (0.0164/2) < 0.05$, $p_{one-tailed-val} = (0.0033/2) < 0.05$).

A.3.2.2 Non-IID Cases

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [79.4, 80.8, 78.4, 79.6, 79.7, 80.8, 77.7, 78.3]
TrimmedMean = [84.9, 86.2, 85.9, 86.6, 84.2, 86.0, 83.2, 84.3]
BARFED = [82.3, 85.6, 85.3, 86.5, 80.8, 83.8, 84.5, 86.3]

$\bar{x}_{CwMedian} = 79.34$, $\bar{x}_{TrimmedMean} = 85.16$, $\bar{x}_{BARFED} = 84.39$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
($\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED}$)
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0015$ < 0.05).

Table 36: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in non-IID setting of Fashion MNIST.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0014	0.0333
TrimmedMean	0.0014	1.0000	0.5715
BARFED	0.0333	0.5715	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.5715 > 0.05$) they both outperform CwMedian ($p_{one_tailed_val} = (0.0014/2) < 0.05$, $p_{one-tailed-val} = (0.0033/2) < 0.05$)

A.4 CIFAR10

A.4.1 Label Flipping Attacks

A.4.1.1 IID Case

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [75.6, 75.7, 73.3, 73.4, 73.8, 73.8, 73.5, 73.6]
TrimmedMean = [75.6, 75.7, 73.3, 73.4, 73.8, 73.8, 73.5, 73.6]
BARFED = [76.2, 76.2, 77.0, 77.2, 77.7, 77.8, 75.6, 75.7]

$\bar{x}_{CwMedian} = 74.09$, $\bar{x}_{TrimmedMean} = 74.09$, $\bar{x}_{BARFED} = 76.68$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

Table 37: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in IID setting of CIFAR10.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.9000	0.0076
TrimmedMean	0.9000	1.0000	0.0076
BARFED	0.0076	0.0076	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and CwMedian ($p_{val} = 0.9000 > 0.05$) but BARFED outperforms CwMedian ($p_{one_{t}ailed_{v}al} = (0.0076/2) < 0.05$) and TrimmedMean ($p_{one_{t}ailed_{v}al} = (0.0076/2) < 0.05$).

A.4.1.2 Non-IID Cases

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [55.0, 55.9, 54.5, 54.9, 56.7, 57.2, 52.0, 52.7]
TrimmedMean = [76.0, 76.0, 72.6, 72.7, 75.4, 75.4, 74.0, 74.1]
BARFED = [78.0, 78.1, 76.8, 76.9, 77.3, 77.4, 76.2, 76.3]

$\bar{x}_{CwMedian} = 54.86$, $\bar{x}_{TrimmedMean} = 74.53$, $\bar{x}_{BARFED} = 77.12$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0003$ < 0.05).

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.1123 > 0.05$) but BARFED outperforms CwMedian ($p_{one-tailed-val} = (0.0010/2) < 0.05$).

Table 38: The pairwise p-values that are obtained from posthoc analysis for label flipping attacks in non-IID setting of CIFAR10.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.1123	0.0010
TrimmedMean	0.1123	1.0000	0.1123
BARFED	0.0010	0.1123	1.0000

A.4.2 Byzantine Attacks

A.4.2.1 IID Cases

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [75.7, 75.8, 75.0, 75.0, 77.6, 77.7, 75.1, 75.1]
TrimmedMean = [78.8, 78.9, 75.6, 75.7, 76.8, 76.9, 77.0, 77.1]
BARFED = [77.8, 77.9, 77.4, 77.5, 77.1, 77.2, 77.3, 77.3]

$\bar{x}_{CwMedian} = 75.88$, $\bar{x}_{TrimmedMean} = 77.10$, $\bar{x}_{BARFED} = 77.44$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

There is no statistically significant difference between performances of CwMedian, TrimmedMean and BARFED according to the result of the the Friedman Chi-Square test ($p_{val} = 0.1353$ > 0.05).

A.4.2.2 Non-IID Cases

The accuracy scores are obtained for each method and the sample mean are presented below.

CwMedian = [57.7, 59.0, 71.8, 72.0, 62.0, 62.3, 58.0, 61.2]
TrimmedMean = [76.4, 76.5, 76.3, 76.4, 76.6, 76.6, 75.5, 75.6]
BARFED = [77.6, 77.6, 77.5, 77.6, 75.2, 75.3, 77.2, 77.2]

$\bar{x}_{CwMedian} = 63.00$, $\bar{x}_{TrimmedMean} = 76.24$, $\bar{x}_{BARFED} = 76.90$

H_0: The performance of CwMedian, TrimmedMean and BARFED are same.
$(\mu_{CwMedian} = \mu_{TrimmedMean} = \mu_{BARFED})$
H_1: At least one of them is different.

The result of the Friedman Chi-Square indicates that the performance of at least one of the methods is statistically different from the others ($p_{val} = 0.0015$ < 0.05).

Table 39: The pairwise p-values that are obtained from posthoc analysis for Byzantine attacks in non-IID setting of CIFAR10.

	CwMedian	**TrimmedMean**	**BARFED**
CwMedian	1.0000	0.0333	0.0014
TrimmedMean	0.0333	1.0000	0.5715
BARFED	0.0014	0.5715	1.0000

The posthoc analysis shows that there is no statistically significant difference between TrimmedMean and BARFED ($p_{val} = 0.5715 > 0.05$) and they both outperforms CwMedian ($p_{one-tailed-val} = (0.0333/2) < 0.05$, $p_{one-tailed-val} = (0.0014/2) < 0.05$).