Federated Learning With Unreliable Clients: Performance Analysis and Mechanism Design

Chuan Ma, *Member, IEEE*, Jun Li, *Senior Member, IEEE*, Ming Ding, *Senior Member, IEEE*, Kang Wei, *Graduate Student Member, IEEE*, Wen Chen, *Senior Member, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*—Owing to the low communication costs and privacy-promoting capabilities, federated learning (FL) has become a promising tool for training effective machine learning models among distributed clients. However, with the distributed architecture, low-quality models could be uploaded to the aggregator server by unreliable clients, leading to a degradation or even a collapse of training. In this article, we model these unreliable behaviors of clients and propose a defensive mechanism to mitigate such a security risk. Specifically, we first investigate the impact on the models caused by unreliable clients by deriving a convergence upper bound on the loss function based on the gradient descent updates. Our bounds reveal that with a fixed amount of total computational resources, there exists an optimal number of local training iterations in terms of convergence performance. We further design a novel defensive mechanism, named deep neural network-based secure aggregation (DeepSA). Our experimental results validate our theoretical analysis. In addition, the effectiveness of DeepSA is verified by comparing with other state-of-the-art defensive mechanisms.**

***Index Terms*—Convergence bound, defensive mechanism, federated learning (FL), unreliable clients.**

I. Introduction

MACHINE learning (ML) technologies, e.g., deep learning, have revolutionized the ways that information is extracted with ground-breaking successes in various areas. Meanwhile, owing to the advent of the Internet of Things (IoT), the number of intelligent applications with edge computing, such as smart manufacturing, intelligent transportation, and intelligent logistics, is growing exponentially [1]–[5]. As such, the conventional centralized deep learning is no longer capable of efficiently processing the dramatically increased amount of data from the massive number of IoT devices. To tackle this challenge, distributed learning frameworks have emerged, e.g., federated learning (FL), enabling the decoupling of data provisioning by distributed clients and aggregating ML models at a centralized server [6]–[8]. Through local training and central aggregating iteratively, FL does not require clients to share their sensitive data with the central server, thereby effectively reducing transmission overheads as well as promoting clients' privacy to some extent [9]–[11].

Although the clients' data are not explicitly exposed in the original format, it is still possible for adversaries to infer clients' private information approximately, especially when the architecture of the FL model and its parameters are not completely protected. Moreover, the existence of unreliable clients may further create security issues in IoT applications. This is because the server in an FL system has no access to the clients' data, nor does it have full control of the clients' behaviors. As a consequence, a client may deviate from the normal behaviors during the course of FL, which is termed an *unreliable client* in this work. Unreliable behaviors may be intentional, e.g., by a malicious attacker disguised as a normal client, or unintentional, e.g., by a client with hardware and/or software limitations/defects in IoT. For example, in smart manufacturing scenarios, engines with sensors that have abnormal traffic and irregular reporting frequency may cause industrial production interruption thus resulting in substantial economic losses for factories [12], [13].

Unreliable clients in FL, for example, may manipulate their outputs sent to the server and they can dominate the training process and change the judging boundary of the global model, or make the global model deviate from the optimal solution. To model these clients, the work in [14] proposed that an unreliable client may interfere with the process of FL by applying limited changes to the uploaded model parameters. The work in [15] proposed a model-replacement method that demonstrated its efficacy on poisoning models of standard FL tasks in IoT. In addition, this work also developed and evaluated a generic constrain-and-scale technique that incorporates the evasion of the defensive mechanism into the abnormal clients' loss function during training. Therefore, how to design defensive algorithms against abnormal clients in FL is of considerable interest. In order to detect abnormal updates in FL, the work of [16] applied the results of client-side cross-validation for reducing the weights of bad updates

Manuscript received January 15, 2021; revised April 15, 2021; accepted May 3, 2021. Date of publication May 12, 2021; date of current version December 7, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61872184, Grant 62002170, and Grant 62071296; in part by the National Key Project under Grant 2020YFB1807700 and Grant 2018YFB1801102; in part by the Sciences and Technology Commission of Shanghai (STCSM) under Grant 20JC1416502; and in part by the U.S. National Science Foundation under Grant CCF-1908308. *(Corresponding author: Jun Li.)*

Chuan Ma, Jun Li, and Kang Wei are with the School of Electrical and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: chuan.ma@njust.edu.cn; jun.li@njust.edu.cn; kang.wei@njust.edu.cn).

Ming Ding is with Data61, CSIRO, Sydney, NSW 2015, Australia (e-mail: ming.ding@data61.csiro.au).

Wen Chen is with the Department of Electronics Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wenchen@sjtu.edu.cn).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Digital Object Identifier 10.1109/JIOT.2021.3079472

TABLE I
SUMMARY OF MAIN NOTATION

Notation	Description
$\boldsymbol{w}_i^{(k\tau)}$	The uploaded model of i-th client at the k-th communication round
τ	The number of local training epochs
$F_i(\cdot)$	The local objective function of the i-th client
$\widehat{\boldsymbol{w}}_i^{(k\tau)}$	The local model of the i-th unreliable client
α	The scaling factor with range [-1,1]
$\boldsymbol{n}$	The additive noise
p_{U}	The probability of the unreliable behavior
p_i	The aggregating weight based on the training data size of $\mathcal{D}_i$
T	The number of total training iterations
k	The number of total aggregation

when performing aggregation, where each update is evaluated over other clients' local data. Similarly, the work in [17] also focused on the weights and they presented a novel aggregation algorithm with the residual-based reweighting method. The work in [18] considered the existence of unreliable participants and utilized auxiliary validation data to compute a utility score for each participant to reduce the impact of these unreliable participants, while the work in [19] directly removed the corresponding model parameters from the training procedure if the accuracy of the client is lower than a predefined threshold. The work in [20] proposed a robust aggregation rule, called adaptive federated averaging, which detects and discards malicious or bad local updates based on a hidden Markov model. The work in [21] performed the first systematic study of local model poisoning attacks on FL, in which they formulate attacks as optimization problems and test four different robust FL methods. However, all of these works lack theoretical analysis on the performance of FL systems in the presence of unreliable clients.

It should be noted that the analysis and optimization for the basic FL system has already been investigated in [22]–[25], yet there are no analytical results on the security aspects in an FL system. Therefore, in this work, we first conduct such an analysis in the context of FL with unreliable clients. We first introduce a model for unreliable clients in FL systems and derive convergence bounds. Through our theoretical results, we find that there exists an optimal local training iteration that leads to a best system performance within a constraint on total computing resources. Then, we design a novel defensive mechanism, referred to as the deep neural network (DNN)-based secure aggregation (DeepSA), to efficiently reduce the negative effects caused by unreliable clients.

The major contributions of this article can be summarized as follows.

1) We involve unreliable clients in FL, in which model parameters will be scaled down and corrupted by noise before uploading. Further, we derive an upper bound on the loss function in FL systems with a given level of computational resources. Our bound reveals that there exists an optimal number of local training epochs to achieve the best convergence performance.
2) We propose a novel defensive mechanism, i.e., DNN-based DeepSA, which can detect abnormal models, and then alleviates the negative impact by removing them from the aggregation.
3) We conduct extensive experiments on the proposed model with the multilayer perceptron (MLP) model and real-life datasets. Our experimental results are shown to be consistent with the theoretical ones. Also, compared with other existing defensive algorithms, the proposed one can improve the FL model performance effectively.

The remainder of this article is organized as follows. Section II introduces the background of FL. Section III details the system models. In Section IV, we analyze the convergence bound of the FL system with unreliable clients. Section V proposes the DeepSA algorithm to address the unreliable problem, and the experimental results are shown in Section VI. Finally, we conclude this article in Section VIII. In addition, a summary of notation is listed in Table I.

II. PRELIMINARIES FOR FEDERATED LEARNING

In this section, we will introduce the basic concepts of FL. As a kind of distributed training frameworks [7], FL can promote user privacy by its unique distributed learning mechanism. In FL, all clients share the same learning objective and model structure, where a central server sends the current global model parameters $\boldsymbol{w}$ to all clients $\mathcal{C}_i$ $\forall i \in \mathcal{M} \triangleq \{1, 2, \ldots, M\}$, in each communication round. Then, all clients update local models based on the shared global model and local data set $\mathcal{D}_i$. After local training, all local models will be uploaded to the server by clients, and then aggregated by the server as the current global model, which is expressed as

$$\boldsymbol{w}^{(k\tau)} = \sum_{i\in\mathcal{M}} p_i \boldsymbol{w}_i^{(k\tau)} \tag{1}$$

where $\boldsymbol{w}_i^{(k\tau)}$ is the uploaded model of the ith client at the kth communication round, $\boldsymbol{w}^{(k\tau)}$ is the global model after aggregation at the kth communication round, τ is the number of local training epochs, and $p_i = |\mathcal{D}_i|/|\mathcal{D}|$ is the aggregating weight based on the size of $\mathcal{D}_i$, where $\mathcal{D} = \sum_{i\in\mathcal{M}} \cup \mathcal{D}_i$ and $|\cdot|$ represents the cardinality of a set, respectively. At the server side, the goal is to learn a model over data that resides at the M associated clients. Formally, this FL task can be expressed as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i\in\mathcal{M}} p_i F_i(\boldsymbol{w}) \tag{2}$$

where $F(\boldsymbol{w}) = \sum_{i\in\mathcal{M}} p_i F_i(\boldsymbol{w})$ and $F_i(\cdot)$ is the local objective function of the ith client.

In addition, the FL in this article adopts the optimization method of gradient descent. In order to capture the divergence between the gradient of a local and global loss function, the gradient divergence is defined as follows [24].

Definition 1: For any $i \in \mathcal{M}$ and $\boldsymbol{w}$, an upper bound on $\|\nabla F_i(\boldsymbol{w}) - \nabla F(\boldsymbol{w})\|$ is defined as δ_i, i.e.,

$$\|\nabla F_i(\boldsymbol{w}) - \nabla F(\boldsymbol{w})\| \leq \delta_i. \tag{3}$$

We also define $\delta \triangleq [(\sum_{i\in\mathcal{M}} |\mathcal{D}_i|\delta_i)/|\mathcal{D}|]$. If the size of each local data set is same, we know that $\delta = (1/M)\sum_{i=1}^{M} \delta_i$.

This divergence is governed by how the data is distributed at different clients.

III. System Models

In FL, unreliable model updates might exist in a wireless transmission environment. Thus, these flawed uploads will impair the effectiveness of the global model, misleading the updated AI model away from optimality. To be more specific, abnormal behaviors generally can be classified into two categories: 1) intentional and 2) unintentional. Intentional adversary clients, also regarded as malicious clients, usually aim to sabotage the system performance or even destroy the learned model. For example, the values of the uploaded parameters may be scaled down or even completely reversed. In contrast, unintentional behaviors could happen without any particular purpose, for example, a noisy version of parameters could be uploaded to the server. In this article, we propose to model both types of abnormal clients.

We consider an FL system consisting of a single central server and M clients, as shown in Fig. 1. We assume that each client may upload unreliable models throughout the whole training process.

A. Adversary Model

Each client is assumed to have a local model with the same structure, and corresponding model parameters uploaded for each epoch are of the same format. The shared model is guided by these parameter vectors to the optimal value. Then, we denote by $\widehat{\boldsymbol{w}}_i^{(k\tau)}$ the local model of the ith unreliable client, and express as

$$\widehat{\boldsymbol{w}}_i^{(k\tau)} = \alpha \boldsymbol{w}_i^{(k\tau)} + \boldsymbol{n}_i^{(k\tau)} \tag{4}$$

where $\alpha \in [-1, 1]$ denotes the scaling factor and $\boldsymbol{n}$ denotes the additive noise and is assumed to follow a Gaussian distribution with $N(0, \sigma^2)$. Equation (4) can well capture the abnormal behaviors as the scalar is used to model the malicious clients and the random noise denotes the undesirable perturbation on the uploaded models. For example, when $\alpha = -1$, it means an adversarial client will completely reverse the uploaded parameters on purpose, and can be recognized as a malicious behavior. Due to the aggregation process in (1), we have

$$\bar{\boldsymbol{w}}^{(k\tau)} = \sum_{i \in M} p_i \left[(1 - p_U) \boldsymbol{w}_i^{(k\tau)} + p_U \hat{\boldsymbol{w}}_i^{(k\tau)} \right] \tag{5}$$

where p_U denotes the probability of the unreliable behavior.[1]

IV. Convergence Analysis

In this section, we will propose a theoretical analysis on the convergence of the FL system considering the existence of abnormal behaviors. For the purpose of facilitating the analysis, we make the following assumptions on the loss function.

Assumption 1: We assume that the following conditions are satisfied for all i $\forall i \in \mathcal{M}$.

1) $F_i(\boldsymbol{w})$ is convex.
2) All model parameters satisfy $\|\boldsymbol{w}\| \leq \Theta$.

[1]We assume a same value of p_U for all clients in this work. Different unreliable probabilities for different clients maybe out of the scope, and can be our future work.

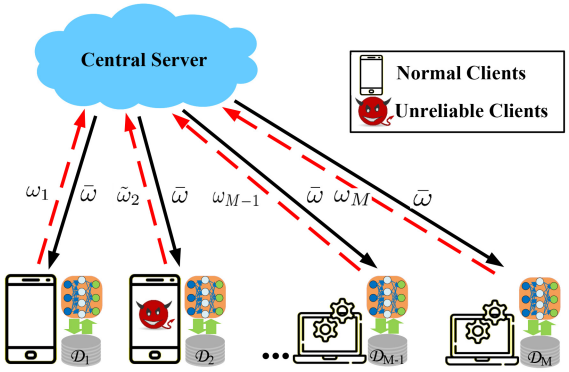


Fig. 1. FL training framework with unreliable clients.

3) $F_i(\boldsymbol{w})$ is ρ-Lipschitz, i.e., $\|F_i(\boldsymbol{w}) - F_i(\boldsymbol{w}')\| \leq \rho \|\boldsymbol{w} - \boldsymbol{w}'\|$, for any $\boldsymbol{w}$, $\boldsymbol{w}'$.
4) $F_i(\boldsymbol{w})$ is β-smooth, i.e., $\|\nabla F_i(\boldsymbol{w}) - \nabla F_i(\boldsymbol{w}')\| \leq \beta \|\boldsymbol{w} - \boldsymbol{w}'\|$, for any $\boldsymbol{w}$, $\boldsymbol{w}'$.
5) $\eta \leq \frac{1}{\beta}$, where η is the step size.
6) $\|F(\boldsymbol{w}^t) - F(\boldsymbol{w}^*)\| \geq \varepsilon$, for all $\boldsymbol{w}$ during FL training.

We also assume that the clients participating in the training hold the same amount of data, i.e., $p_i = (1/M)$. In general, these assumptions with some restrictions for the convenience of theoretical derivation can be satisfied.

A. Convergence Analysis

In this section, we evaluate the performance of FL under abnormal behaviors by an upper bound on the difference between $\mathbb{E}\{F(\bar{\boldsymbol{w}}^{(T)})\}$ and $F(\boldsymbol{w}^*)$, where $\bar{\boldsymbol{w}}^{(T)}$ is the final global parameters of the FL system containing M potential unreliable clients and $\boldsymbol{w}^*$ is the optimal model parameters that minimizes $F(\boldsymbol{w})$.

Theorem 1: For some $\varepsilon > 0$ and $\Theta > 0$, when the clients in the FL system behave unreliably with probability p_U, the convergence upper bound with a fixed total number of iterations T is given by

$$\mathbb{E}\left\{F\left(\bar{\boldsymbol{w}}^{(T)}\right)\right\} - F\left(\boldsymbol{w}^*\right) \leq \frac{1}{T\left(\omega\eta\left(1 - \frac{\beta\eta}{2}\right) - \frac{\rho\left(\phi(\tau) + \frac{p_U}{M}\left[(1-\alpha)M\Theta + \frac{2\sqrt{M}\sigma}{\pi}\right]\right)}{\tau\varepsilon^2}\right)} \tag{6}$$

where $\phi(\tau) = (\delta/\beta)((\eta\beta + 1)^\tau - 1) - \eta\delta\tau$, $\varphi = \omega(1 - [\beta\eta/2])$, and $\omega \triangleq \min_k [1/(\|\boldsymbol{w}^{(k\tau)} - \boldsymbol{w}^*\|^2)]$.

Proof: See Appendix A. ■

The upper bound given by Theorem 1 demonstrates the convergence result of the FL system with unreliable clients. A lower bound means that the value of the system loss function converges closer to the optimal one.

B. Discussion of the Convergence Bound

In this section, we will provide several key observations about the convergence bound.

Proposition 1: If there is no unreliable client, the convergence bound of FL increases, which also means a *worse* system performance, as the local epochs τ increases. Since

other parameters are basically fixed, the influence of τ on this theoretical value is the most noteworthy.

Proof: When $p_U = 0$, the convergence upper bound can be expressed as $1/T(\omega\eta(1 - [\beta\eta/2]) - [(\rho(\phi(\tau)))/(\tau\varepsilon^2)])$. It is evident that $[(\rho(\phi(\tau)))/(\tau\varepsilon^2)]$ increases with a larger τ and leads to a larger bound. ■

Proposition 2: When the probability of unreliable behaviors p_U is larger, the convergence performance becomes worse. However, when this percentage is fixed, the performance of the system will *improve* with the number of total clients M.

Proof: Considering the analytical part in the convergence bound related to M, we find the value $(p_U/M)[(1-\alpha)M\Theta + [(2\sqrt{M}\sigma)/\pi]]$ decreases with the increase of M when p_U is fixed, thus, the convergence bound becomes smaller. ■

We note that when the total number of iterations T is constant, the local training iterations τ should be as small as possible if there is no unreliable client. This is because when $\tau = 1$, the FL system based on distributed gradient descent is equivalent to a centralized training model [24]. However, in an unreliable circumstance, there exists an optimal value of $\tau \in [1, T]$ (T is an integer multiple of τ) to have the optimal convergence performance. Therefore, we can make the following proposition.

Proposition 3: Under the unreliable behaviors of clients with a fixed T, the convergence upper bound is a convex function of the number of local epochs τ, if we treat τ as a continuous variable.

Proof: See Appendix B. ■

From Proposition 3, we can see that there exists an optimal τ which can minimize the value of the loss function to obtain a satisfactory learning performance.

V. Defensive Mechanism Design

In this section, we will use a crafted DNN to detect the existence of unreliable clients. Current defensive mechanisms, such as Secprobe [18] and Krum [26], usually need an online testing data set to adjust the aggregation weight. The testing data set is either from clients, which may pose privacy issue, or using a public one that may affect the accuracy. Thus, in this work, we consider training an offline detector to recognize the abnormal clients. A basic binary anomaly detection technique using DNN operates in two steps. First, the DNN is trained on the normal training data to learn all normal labels. Second, each test instance is provided as an input to the DNN. If the DNN accepts the test input, it is labeled as normal and if the network rejects a test input, it is an anomaly. Therefore, we propose the DeepSA algorithm based on a crafted DNN in a one-class setting. The main implemental process is operated in the server with a new functional module. To complete this module, the detector is pretrained before FL with several normal parameter inputs, and these parameters can be obtained from clients or a public data set. Once the pretraining process ends, this module can be used for anomaly detection.

In FL, the set of local models received by the server at the kth communication round can be expressed as

$$\boldsymbol{o}^{(k)} \triangleq \left\{ \boldsymbol{w}_1^{(k\tau)}, \boldsymbol{w}_2^{(k\tau)}, \ldots, \boldsymbol{w}_i^{(k\tau)}, \ldots, \boldsymbol{w}_U^{(k\tau)} \right\}. \tag{7}$$

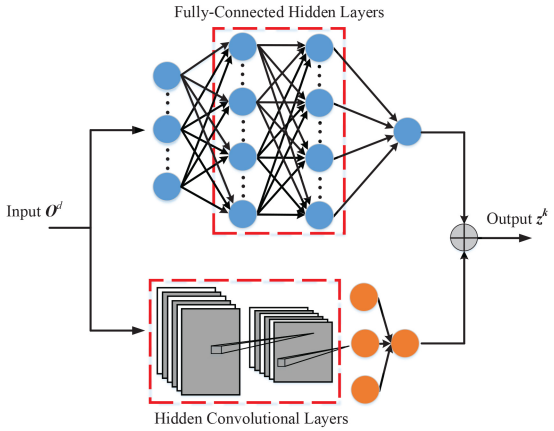


Fig. 2. Architecture of the DNN-based detector.

We use a DNN-based anomaly detector, denoted by $\mathfrak{D}$, which can be viewed as a classifier to assign a label (normal or abnormal). Typically, the outputs produced by this detector, are one of the following two types: 1) *Scores*: scoring techniques assign a detecting score to each instance, which is utilized to analyze the possibility of unreliable clients and 2) *Labels*: techniques in this category assign a label (benign or malicious) to each test instance. In our DNN-based detector, we define the detecting result of a test instance (or observation) by $\boldsymbol{z}^{(k)} = [z_1^{(k)}, z_2^{(k)}, \ldots, z_U^{(k)}] \in \{0, 1\}^U$. If $z_i = 0$, it represents that $\boldsymbol{w}_i^{(k\tau)}$ is the unreliable model. Therefore, the detecting process can be given by

$$\boldsymbol{z}^{(k)} = \mathfrak{D}\left(\boldsymbol{o}^{(k)}, \boldsymbol{o}^{(k-1)}, \ldots, \boldsymbol{o}^{(1)}\right) \tag{8}$$

where $\mathfrak{D}$ is the detector and we assume that whole sets of local models (from 1st to kth communication round) can be utilized as the input of this detector.

In detail, the server will receive the observations $\boldsymbol{o}^{(k)}$ at the kth communication round, and try to identify these abnormal models in them. For a normal client, there should be a certain level of correlation among its uploaded parameters in consecutive communication rounds. However, manipulation of parameters by anomaly clients may break this correlation, hence, the previous observations can also assist in detecting abnormal models. Therefore, in order to enhance this difference, we use an input reshaping approach to ensure the shift-invariance property for our detector, which can be expressed as

$$\boldsymbol{O}^d = \begin{bmatrix} \boldsymbol{w}_1^{(k\tau)} & \boldsymbol{w}_1^{((k-1)\tau)} & \cdots & \boldsymbol{w}_1^{((k-d)\tau)} \\ \boldsymbol{w}_2^{(k\tau)} & \boldsymbol{w}_2^{((k-1)\tau)} & \cdots & \boldsymbol{w}_2^{((k-d)\tau)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{w}_{U-1}^{(k\tau)} & \boldsymbol{w}_{U-1}^{((k-1)\tau)} & \cdots & \boldsymbol{w}_{U-1}^{((k-d)\tau)} \\ \boldsymbol{w}_U^{(k\tau)} & \boldsymbol{w}_U^{((k-1)\tau)} & \cdots & \boldsymbol{w}_U^{((k-d)\tau)} \end{bmatrix} \tag{9}$$

where d is the depth of the observation. The input $\boldsymbol{O}^d$ is shaped as a multidimensional vector ($U \times 1 \times d \times s_w$), where s_w is the size of the standard uploaded model. With this input design, we will introduce the construction of our DNN-based detector.

As shown in Fig. 2, the DNN detector consists of two parallel pipelines and achieves an output with the XOR operator.

Algorithm 1 Secure Aggregation on the Server Side

Input: The observation of all uploaded models with d depth $\boldsymbol{O}^d$, the well trained DNN based detector $\mathfrak{D}$
Output: The global model $\boldsymbol{w}^{(k\tau)}$
1: Wait for clients to upload their weights until there are M clients' models $\boldsymbol{o}^k = \left\{\boldsymbol{w}_1^{(k\tau)}, \boldsymbol{w}_2^{(k\tau)}, \ldots, \boldsymbol{w}_i^{(k\tau)}, \ldots, \boldsymbol{w}_M^{(k\tau)}\right\}$
2: Update the input $\boldsymbol{O}^d$ with the fresh uploaded models $\boldsymbol{o}^k$
3: Obtain the detecting results with the DNN based detector $\mathfrak{D}$ as $\boldsymbol{z}^k = \mathfrak{D}\left(\boldsymbol{O}^d\right)$
4: Average the benign models and obtain $\boldsymbol{w}^{(k\tau)} = \sum_{i=1}^{M} z_i p_i \boldsymbol{w}_i^{(k\tau)}$
5: Send the new averaged model $\boldsymbol{w}^{(k\tau)}$ to all clients
6: **return** $\boldsymbol{w}^{(k\tau)}$

The input is $\boldsymbol{O}^k$ and the output is the symbol $\boldsymbol{z}^k$. Using the fully connected layers and sufficient training, we can identify the intentional behaviors, such as the scaling operation on the parameters. In addition, for the randomized parameters, the convolutional layer can be more useful since the correlation between the noised and normal parameters is expected to be low. In the convolutional layer, we use zero padding with stride size B, and set the filter size to $B \times l$, where l is the depth of the filter. This setting is based on the observation that each subvector is strongly correlated with $2B$ neighboring subvectors due to the structure of uploaded models.

After introducing this crafted DNN-based detector, we present our proposed defence algorithm as shown in Algorithm 1. As discussed above, the existence of abnormal clients indicates that the parameters uploaded by them may be disruptive, and it may reduce the accuracy of the global model. To mitigate their effect on the model accuracy, we remove the malicious models in this communication rounds using our DNN-based detector.

Algorithm 1 gives the pseudocode of secure aggregation on the server side. The server first waits for the local model from each client. When all the clients finish uploading their models to the server, these models are utilized to update the input $\boldsymbol{O}^d$ with the fresh uploaded models. Then, the server can obtain the detecting results with the DNN-based detector and average the benign models. For the whole system, reducing the number of clients is equivalent to reducing the amount of training data, which will reduce the generalization of the global model. However, compared with the damage brought by unreliable clients, these losses are acceptable. In addition, considering that even a reliable client may upload a model with poor quality, the decision result of the DNN detector will only take effect in the current round of communication.

VI. Experimental Results

In this section, we first evaluate the performance of the analytical results with unreliable clients and verify the effectiveness with the experimental results. Then, we demonstrate the effectiveness of the proposed defensive mechanism by comparing with other algorithms.[2]

[2]Related codes can be found in the following link: https://github.com/JJisbug/UnreliableClientsinFL.

A. Experimental Settings

1) Data Set: In our experimental results, we use four benchmark data sets for different tasks.

1) MNIST and Fashion-MNIST data sets, which both have $70K$ digit images of size 28×28, are split into $60K$ training and $10K$ test samples.
2) The CIFAR-10 data set, which consists of $60K$ color images in ten object classes, such as deer, airplane, and dog with 6000 images included per class, is split into $50K$ training and $10K$ test samples.
3) The Adult data set, which has around $32K$ tabular samples and each sample has 14 attributes, is split into $20K$ training and $12K$ test samples.

We consider the data as being independent identically distributed (i.i.d.), i.e., clients in the FL system possess the same amount of data from training sets randomly and independently.

2) Parameter Settings: We use the MLP as the training model to construct the FL system, and each client locally computes stochastic gradient descent (SGD) updates on each data set, and then aggregate updates to train a globally shared classifier. We conduct three cases for the unreliable client to verify the analytical results as follows.

1) *Case I:* $\alpha = -1$ and $\sigma = 0.1$, in which an unreliable client uploads a completely inverse parameter with small noise.
2) *Case II:* $\alpha = 0.8$ and $\sigma = 0.5$, in which a large noise is added.
3) *Case III:* $\alpha = 0.5$ and $\sigma = 0.3$, in which the parameter is scaling half with a medium noise.

In addition, we set the total number of clients $M = 50$ and the total learning iterations $T(k\tau) = 300$. We run each experiment for 20 times and record the average results. If a client uploads unreliable parameters in all communication rounds, then this scenario can be treated as a special case in which we assume that there are certain percentages of unreliable clients, and other clients will upload reliable parameters during the whole training process.

3) Comparing Defensive Mechanisms: To show the effectiveness of the proposed defensive mechanisms, we provide the following defensive mechanisms.

1) *Krum [26]:* The aggregated parameters are chosen according to the minimum geometric gradient rule.
2) *Secprobe [18]:* The aggregated parameters are chosen according to the testing accuracy.
3) *Pearson [27]:* The aggregated weights are adjusted by the Pearson correlation.

B. Theoretical Results

In Fig. 3, we show the experimental results of loss function value as a function of τ with $p_U = 0.1$ under the unreliable environment. In order to be close to reality [the local training epoch (τ) and communication rounds (k) of clients are not too small], the range of τ is set to $[10, 100]$. We can observe that the theoretical bounds are convex functions and close to the real results for the three cases and four data sets, which are consistent with Theorem 1 and Proposition 3. The reason behind this phenomenon is that a large local epochs

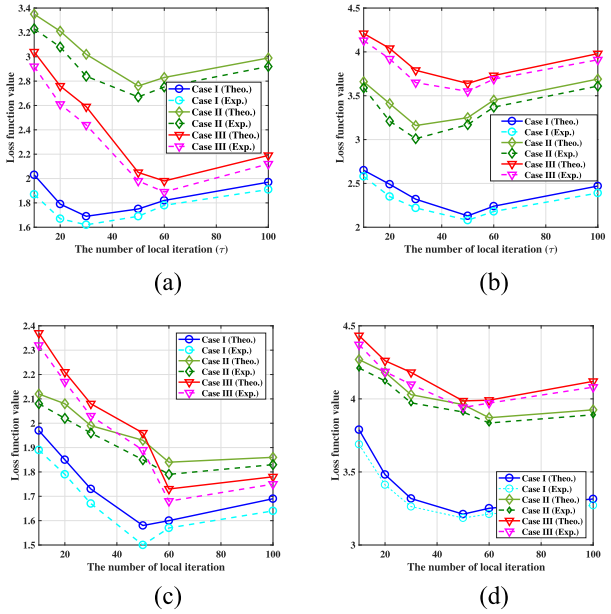


Fig. 3. Comparison of the loss function value between the theoretical and experimental results. (a) MNIST data set. (b) Fashion-MNIST data set. (c) Cifar-10 data set. (d) Adult data set.

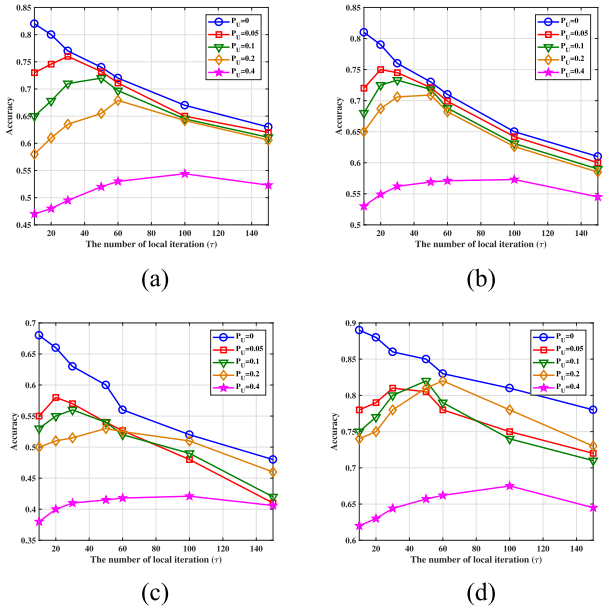


Fig. 4. Classification accuracy of local iterations with a certain probability of unreliable behavior. (a) MNIST data set. (b) Fashion-MNIST data set. (c) Cifar-10 data set. (d) Adult data set.

τ will decrease the times of uploading unreliable parameters, while a small τ incurs much unreliability in the parameters uploaded by all clients. In addition, with a smaller value of added noise, the learning performance will get fewer negative influences.

C. Experimental Results With Unreliable Behavior

In this section, we show the classification accuracy based on the FL system with different probabilities of unreliable clients in Fig. 4. We take case I as the abnormal client for MNIST and Fashion-MNIST data sets, case II for the Cifar-10 data set, and case III for the Adult data set, respectively. In order to show different conditions, we also set the probability of unreliable clients p_U to 0.05, 0.1, 0.2, and 0.4, respectively. From these figures, we find that when there is no abnormal client ($p_U = 0$), the system performance decreases with the increase of the local iteration τ, which is consistent with Proposition 1. However, when the uploading environment is unreliable, we can note that there exists an optimal number of local iterations τ in terms of system performance, which is in line with Proposition 3. We can also note that the optimal number of local training iterations increases with the probability of abnormal clients. The intuition is that more communication rounds will produce a larger damage to the FL system, but more communication rounds also bring a better performance for a normal FL system. In addition, we find that, with an increasing probability of unreliable probability, the system performance shows a descending trend, and a system with relatively high probability, i.e., $p_U > 50\%$, may fall to converge.

In Fig. 5, we show the loss function value under different numbers of total clients that we set M to 50, 100, 150, 200, and 250. In Fig. 5(a), we use case I, in Fig. 5(b) and (c), we use case II, and in Fig. 5(d), we use case III, respectively.

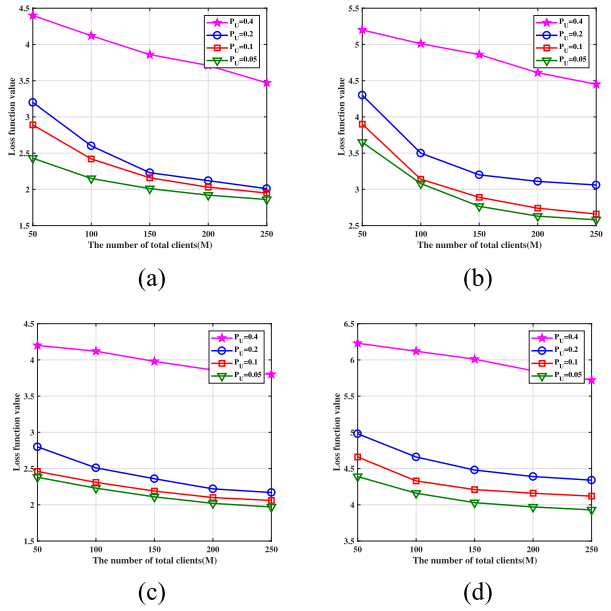


Fig. 5. Loss function value with different number of total clients. (a) MNIST data set. (b) Fashion-MNIST data set. (c) Cifar-10 data set. (d) Adult data set.

We can note that the system has a better performance with a smaller unreliable probability (p_U), and the loss function value in both figures keeps decreasing with the number of total clients, which is consistent with Proposition 2.

D. Performance of the Proposed DeepSA Algorithm

In this section, we conduct experiments on our proposed DeepSA-based federated training against various percentages of unreliable clients. We use ReLU as the activation functions for the hidden layers: $y = \mathrm{ReLU}(x) = \max(x, 0)$, where $x \in \mathbb{R}$ is the input, and y is the output of the activation function. To map the output to the interval between $(0, 1)$, we

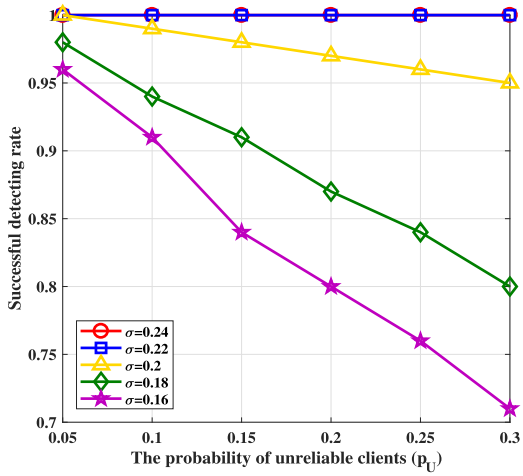


Fig. 6. Successful detecting rate with a trained DNN detector under a federated model against different probabilities of unreliable clients p_U.

TABLE II
CLASSIFICATION ACCURACY COMPARISON IN THE MNIST/FASHION MNIST DATA SET WITH DIFFERENT UNRELIABLE PROBABILITIES

	All reliable	Deep-SA	Krum	Secprobe	Pearson
Case I	0.88/0.85	**0.81/0.78**	0.71/0.69	0.78/0.75	0.74/0.71
Case II	0.88/0.85	**0.8/0.765**	0.72/0.69	0.77/0.74	0.76/0.72
Case III	0.88/0.85	**0.81/0.77**	0.73/0.7	0.78/0.75	0.74/0.7

choose the sigmoid function as the activation function for the output layer: $y = \text{sigmoid}(x) = [1/(1 + e^{-x})]$. In Fig. 6, we show the detecting results with a stable scalar $\alpha = 0.8$ and various noises. From this figure, we can observe that with a larger standard deviation σ, the trained DNN-based detector will perform better in identifying these unreliable clients. This is because with a larger standard deviation, a more obvious difference of parameters from neighboring communication rounds will be recognized by the trained detector. Moreover, it can be noted that when $\sigma > 0.22$, this detector will have an excellent performance, which can guarantee no errors, i.e., the detecting rate is 1. We can also find that if p_U is larger, the successful detecting rate will decrease which means that it is more difficult for detectors to identify.

Fig. 7 show the comparison results between the proposed DeepSA algorithm and others, in which we set $p_U = 0.2$ and unreliable behaviors consider cases I–III. In addition, in Tables II and III, we consider a more practical scenario in which clients may behave unreliably with different probabilities. In detail, we assume that there are 100 clients which are divided into four equal-size groups. The probabilities of unreliable behaviors for the four groups are set to 0.1, 0.2, 0.3, and 0.4, respectively. It is obvious that with our proposed DeepSA algorithm, the federated training process performs better in most cases. The reason is that the similarity-based detection algorithms (Secprobe and Pearson) can only handle the noise perturbation behavior, and Krum loses its performance due to the limited number of participants, while the proposed algorithm has a high detecting rate which enhances the learning performance.

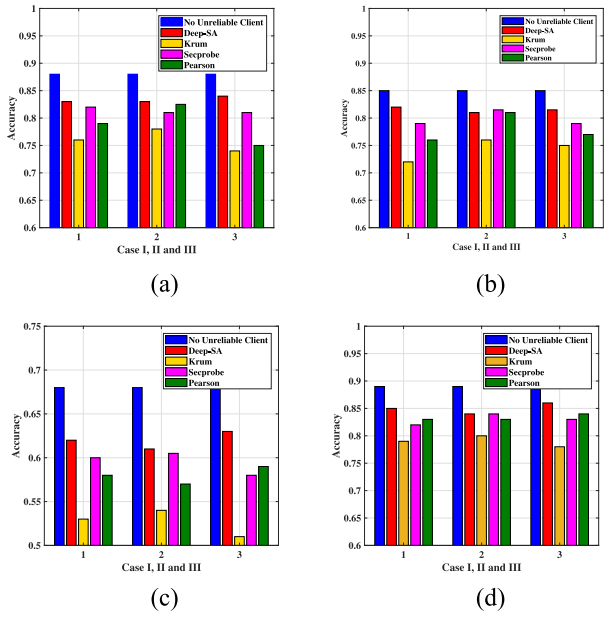


Fig. 7. Classification accuracy comparison between the proposed Deep-SA algorithm and others. (a) MNIST data set. (b) Fashion-MNIST data set. (c) Cifar-10 data set. (d) Adult data set.

TABLE III
CLASSIFICATION ACCURACY COMPARISON IN THE CIFAR-10/ADULT DATA SET WITH DIFFERENT UNRELIABLE PROBABILITIES

	All reliable	Deep-SA	Krum	Secprobe	Pearson
Case I	0.68/0.89	**0.58/0.82**	0.51/0.77	0.57/0.8	0.54/0.78
Case II	0.68/0.89	**0.59/0.83**	0.52/0.79	0.57/0.81	0.56/0.79
Case III	0.68/0.89	**0.6/0.83**	0.52/0.78	0.58/0.81	0.54/0.78

In addition, the proposed defensive mechanism is applied to four real-world data sets, i.e., Sports,[3] UAV Detection,[4] Energy,[5] and Space Shuttle,[6] which have been collected from real-life sensors [28], [29], and the descriptions of these data sets are listed as follows.

1) *Sports:* This data set comprises motion sensor data of 19 daily and sports activities each performed by eight subjects in their own style for 5 min, and we evaluate the performance by the accuracy of a 19-class classifier.
2) *UAV Detection:* This data set consists of 55 attributes in which each data row represents an encrypted WiFi traffic record. The output shows the current traffic is from a UAV or not.
3) *Energy:* This data set consists of assessing the heating load and cooling load requirements of buildings (i.e., energy efficiency) as a function of building parameters with eight attributes, and aims to predict each of two responses.

[3]https://archive.ics.uci.edu/ml/datasets/Daily+and+Sports+Activities
[4]https://archive.ics.uci.edu/ml/datasets/Unmanned+Aerial+Vehicle+%28UAV%29+Intrusion+Detection
[5]https://archive.ics.uci.edu/ml/datasets/Energy+efficiency
[6]https://archive.ics.uci.edu/ml/datasets/Statlog+(Shuttle)

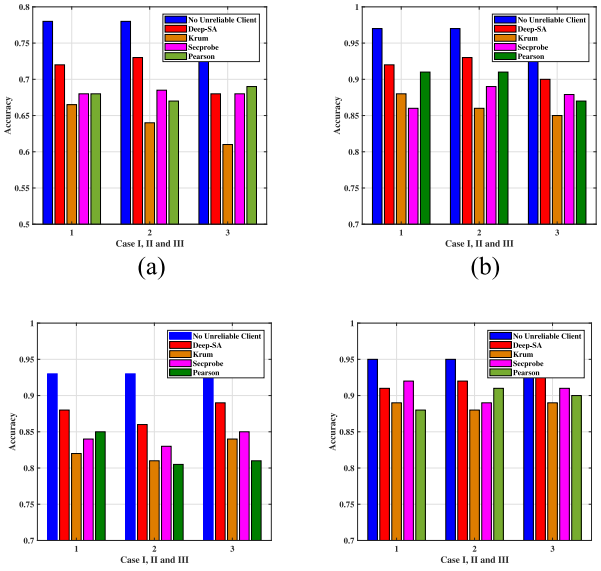


Fig. 8. Classification accuracy comparison between the proposed Deep-SA algorithm and others. (a) Sports data set. (b) UAV Detection data set. (c) Energy data set. (d) Space Shuttle data set.

4) *Space Shuttle:* This shuttle data set contains nine attributes and 58 000 numerical instances with an 80% default accuracy.

In Fig. 8, we verify the proposed defensive algorithm under cases I–III of unreliable clients. As can be found in this figure, although some algorithms may have a slightly better or equal performance than the proposed algorithm, Deep-SA outperforms other algorithms in most cases. For example, in the Sports data set, although DeepSA has not achieved the top performance in case III (68% versus 69%), it has a performance gain in other cases (4% in case I and 4.5% in case II, respectively).

VII. Related Works

In this section, we investigate different adversarial models in FL and defensive mechanisms, which are active areas of research.

A. Adversarial Models in Federated Learning

The security of ML has attracted heated attention recently [30], [31]. Although the data are not explicitly exposed in the original format in distributed learning frameworks, e.g., FL [10], different types of adversarial models against distributed ML algorithms have been designed and analyzed including poisoning attacks (e.g., [21] and [32]) and privacy attacks (e.g., [33]–[35]). For example, poisoning attackers can control part of clients and manipulate the outputs sent to the server, which can mislead the global model deviate to the designed direction [21]. Baruch *et al.* [14] proposed a novel attacking method that a malicious opponent may interfere with the learning process by applying limited changes to the uploaded models. In addition, Bhagoji *et al.* [32] explored the adversarial of model poisoning attacks on FL, which supported by a single, noncolluding malicious client where the adversarial objective is to make the global model misclassify a set of chosen inputs with high confidence.

B. Defensive Mechanisms

With the development of adversarial models in FL, how to design an effective defensive mechanism to defeat these malicious clients has become crucial. For detecting poisoned updates in the collaborative learning [16], the results of client-side cross-validation were applied for adjusting the weights of the updates when performing aggregation, where each update is evaluated over other clients' local data. A similar approach based on Pearson similarity is proposed in [27]. In addition, Zhao *et al.* [18], [36] considered the existence of unreliable clients in Fl and used the auxiliary validation data to compute a utility score for each participant, thus reducing the negative impact of these unreliable participants. The work in [26] proposed a novel poisoning defensive method in Fl. In detail, for each client, the server will calculate the sum of the Euclidean distances to the models of other clients, and select the one with the minimum sum. However, the mentioned defensive algorithms all need an online detection process while the access to the auxiliary data set may leak privacy.

VIII. Conclusion

In this work, we have introduced a new threat model of adversary clients for FL systems. By deriving a convergence bound on the loss function of the trained FL model, we have seen that there exists an optimal number of local training iterations to achieve the best performance with a fixed total amount of computing resources. Furthermore, we have designed a novel defensive algorithm using the DNN detection technique, termed DeepSA, which can automatically detect unreliable models and remove them from the aggregation process. Extensive experimental results have validated our analysis and the effectiveness of the proposed DeepSA algorithm.

Appendix A
Proof of Theorem 1

When we consider the unreliable behavior and (5), we have

$$\bar{\boldsymbol{w}}^{(k\tau)} = \frac{1}{M}\sum_{i\in M}\left[(1-p_U)\boldsymbol{w}_i + p_U\hat{\boldsymbol{w}}_i\right]. \tag{10}$$

Then, according to Assumption 1, the difference between $F(\bar{\boldsymbol{w}}(k\tau))$ and $F(\boldsymbol{w}(k\tau))$ can be expressed as

$$\begin{aligned} F\left(\bar{\boldsymbol{w}}^{(k\tau)}\right) - F\left(\boldsymbol{w}^{(k\tau)}\right) &\le \rho\left\|\bar{\boldsymbol{w}}^{(k\tau)} - \boldsymbol{w}^{(k\tau)}\right\| \\ &= \frac{\rho}{M}\left\|\sum_{i\in M} p_U\left(\hat{\boldsymbol{w}}_i - \boldsymbol{w}_i\right)\right\| \\ &= \frac{\rho}{M}\left\|\sum_{i\in M} p_U[(\alpha-1)\boldsymbol{w}_i + \boldsymbol{n}_i]\right\| \end{aligned}$$

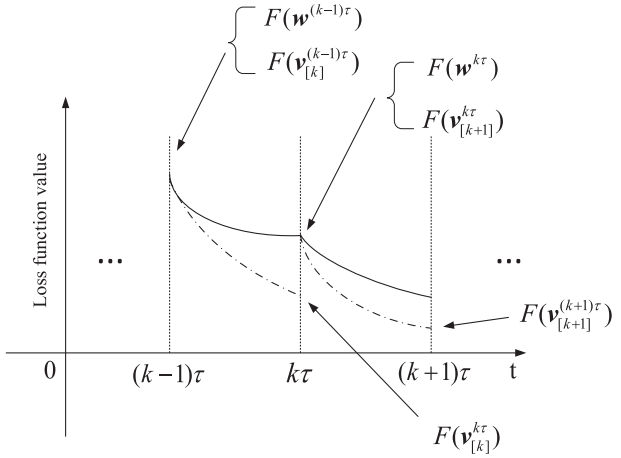


Fig. 9. Auxiliary parameter vector in FL.

$$\leq \frac{\rho p_U}{M}\left[\mathbb{E}\left\|\sum_{i\in M}(\alpha-1)\boldsymbol{w}_i\right\| + \mathbb{E}\left\|\sum_{i\in M}\boldsymbol{n}_i\right\|\right] \leq \frac{\rho p_U}{M}\left[(1-\alpha)M\Theta + \frac{2\sqrt{M}\sigma}{\pi}\right] \quad (11)$$

where $\|\|$ denotes the L_2 norm function, and Θ is an upper bound of all model parameters, respectively. For simplicity, we omit the superscript $(k\tau)$ of $\boldsymbol{w}_i$ and $\boldsymbol{n}_i$.

We define $\boldsymbol{v}_{[k]}^{(t)}$ as an auxiliary parameter vector, which follows a centralized gradient decent:

$$\boldsymbol{v}_{[k]}^{(t)} = \boldsymbol{v}_{[k]}^{(t-1)} - \eta\nabla F\left(\boldsymbol{v}_{[k]}^{(t-1)}\right). \quad (12)$$

The notation $[k]$ defines the interval $[(k-1)\tau, k\tau]$ for $k = 1, 2, \ldots, K$, and the auxiliary parameter vector $\boldsymbol{v}_{[k]}^{(t)}$ only works in this interval. We denote by $\boldsymbol{v}_{[k]}^{(k\tau)}$ and $\boldsymbol{v}_{[k+1]}^{(k\tau)}$ the two different auxiliary vectors before and after the kth aggregation, respectively, which can be seen in Fig. 9. At the beginning of the interval $[k]$, $\boldsymbol{v}_{[k]}^{(t)}$ will not inherit the last result of the previous interval but is equivalent to the global parameter after kth aggregation, i.e., $\boldsymbol{v}_{[k+1]}^{(k\tau)} \triangleq \boldsymbol{w}^{(k\tau)}$.

Using the upper bound in [24], we know that

$$F\left(\boldsymbol{w}^{(k\tau)}\right) - F\left(\boldsymbol{v}_{[k]}^{(k\tau)}\right) \leq \rho\phi(\tau) \quad (13)$$

where k is the index of the aggregation, and $\phi(\tau) = (\delta/\beta)((\eta\beta+1)^\tau - 1) - \eta\delta\tau$.

Combining (11) with (13), we can obtain

$$F\left(\bar{\boldsymbol{w}}^{(k\tau)}\right) - F\left(\boldsymbol{v}_{[k]}^{(k\tau)}\right) \leq \rho\phi(\tau) + \frac{\rho p_U}{M}\left[(1-\alpha)M\Theta + \frac{2\sqrt{M}\sigma}{\pi}\right] = \rho\left\{\phi(\tau) + \frac{p_U}{M}\left[(1-\alpha)M\Theta + \frac{2\sqrt{M}\sigma}{\pi}\right]\right\} = \rho\Delta \quad (14)$$

where $\phi(\tau)+[p_U/M][(1-\alpha)M\Theta + [(2\sqrt{M}\sigma)/\pi]]$ is denoted by Δ. Then, we define $\theta_{[k]}^{(t)} = F(\boldsymbol{v}_{[k]}^{(t)}) - F(\boldsymbol{w}^*)$ for an interval $[k]$, where k is fixed, t is defined between $(k-1)\tau \leq t \leq k\tau$. According to Assumption 1 and [24], we have

$$\theta_{[k]}^{(t)} > \varepsilon \quad (15)$$

and

$$\frac{1}{\theta_{[k]}^{(T)}} - \frac{1}{\theta_{[1]}^{(0)}} \geq \sum_{k=1}^{K-1}\left(\frac{1}{\theta_{[k+1]}^{(k\tau)}} - \frac{1}{\theta_{[k]}^{(k\tau)}}\right) + T\omega\eta\left(1 - \frac{\beta\eta}{2}\right). \quad (16)$$

According to the definition of $\boldsymbol{v}_{[k+1]}^{(k\tau)}$, we have

$$\boldsymbol{v}_{[k+1]}^{(k\tau)} = \bar{\boldsymbol{w}}^{(k\tau)}. \quad (17)$$

Then, we have

$$\frac{1}{\theta_{[k+1]}^{(k\tau)}} - \frac{1}{\theta_{[k]}^{(k\tau)}} = \frac{\theta_{[k]}^{(k\tau)} - \theta_{[k+1]}^{(k\tau)}}{\theta_{[k]}^{(k\tau)}\theta_{[k+1]}^{(k\tau)}} = \frac{F\left(\boldsymbol{v}_{[k]}^{(k\tau)}\right) - F\left(\boldsymbol{v}_{[k+1]}^{(k\tau)}\right)}{\theta_{[k]}^{(k\tau)}\theta_{[k+1]}^{(k\tau)}} = \frac{F\left(\boldsymbol{v}_{[k]}^{(k\tau)}\right) - F\left(\bar{\boldsymbol{w}}^{(k\tau)}\right)}{\theta_{[k]}^{(k\tau)}\theta_{[k+1]}^{(k\tau)}} \geq -\frac{\rho\Delta}{\varepsilon^2}. \quad (18)$$

Combining (16) with (18), we have

$$\frac{1}{\theta_{[k]}^{(T)}} - \frac{1}{\theta_{[1]}^{(0)}} \geq -\frac{\rho(K-1)\Delta}{\varepsilon^2} + T\omega\eta\left(1 - \frac{\beta\eta}{2}\right). \quad (19)$$

When $k = K$, according to Assumption 1, we can obtain

$$F\left(\widehat{\boldsymbol{w}}^{(T)}\right) - F\left(\boldsymbol{w}^*\right) \geq \varepsilon. \quad (20)$$

Therefore, (20) can be expressed as

$$\frac{1}{F\left(\bar{\boldsymbol{w}}^{(T)}\right) - F(\boldsymbol{w}^*)} - \frac{1}{\theta_{[K]}^{(T)}} = \frac{\theta_{[K]}^{(T)} + F(\boldsymbol{w}^*) - F\left(\bar{\boldsymbol{w}}^{(T)}\right)}{\left(F(\bar{\boldsymbol{w}}^{(T)}) - F(\boldsymbol{w}^*)\right)\theta_{[K]}^{(T)}} = \frac{F\left(\boldsymbol{v}_{[k]}^{(T)}\right) - F\left(\bar{\boldsymbol{w}}^{(T)}\right)}{\left(F\left(\bar{\boldsymbol{w}}^{(T)}\right) - F(\boldsymbol{w}^*)\right)\theta_{[K]}^{(T)}} \geq \frac{-\rho\Delta}{\varepsilon^2}. \quad (21)$$

Summing up (19) and (21), we have

$$\frac{1}{F\left(\bar{\boldsymbol{w}}^{(T)}\right) - F(\boldsymbol{w}^*)} - \frac{1}{\theta_{[1]}^{(0)}} = T\omega\eta\left(1 - \frac{\beta\eta}{2}\right) - \frac{\rho K\Delta}{\varepsilon^2} \overset{(a)}{=} T\left(\omega\eta\left(1 - \frac{\beta\eta}{2}\right) - \frac{\rho\Delta}{\tau\varepsilon^2}\right) \quad (22)$$

where step (a) is obtained by $T = K\tau$.

Note that $\theta_{[1]}^{(0)} > 0$, the above inequality can be simplified as

$$\frac{1}{F\left(\widehat{\boldsymbol{w}}^{(T)}\right) - F(\boldsymbol{w}^*)} \geq T\left(\omega\eta(1 - \frac{\beta\eta}{2}) - \frac{\rho\Delta}{\tau\varepsilon^2}\right). \quad (23)$$

According to the definition, we know that $\boldsymbol{w}^*$ is the optimal model parameters minimizing $F(\boldsymbol{w})$. Hence, $F(\widehat{\boldsymbol{w}}_{[k]}^{(T)}) \geq F(\boldsymbol{w}^*)$ and this inequality can be true when the right-hand side of the inequality is less than 0. Note that when the upper bound

on the parameter Θ or the unreliable p_U or the additive noise power σ is big enough,

$$T\left(\omega\eta\left(1-\frac{\beta\eta}{2}\right)-\frac{\rho\left(\phi(\tau)+\frac{p_U}{M}\left[(1-\alpha)M\Theta+\frac{2\sqrt{M}\sigma}{\pi}\right]\right)}{\tau\varepsilon^2}\right)$$

will less than zero. In this case, although the inequality (23) is true or not true it will make no any sense. This can be interpreted as that the system will crash when the learning circumstance is unacceptable. Similarly, when local training epoch τ continues to increase without limitation, the right-hand side of this inequality will be smaller than zero. Therefore, for ease of analysis, we assume that there are certain limits on Θ, p_U, σ, and τ. In other words, we assume that $T(\omega\eta(1-[\beta\eta/2])-[\rho\Delta/\tau\varepsilon^2])>0$. Then, taking the reciprocal of the above inequality yields

$$F\left(\widehat{\boldsymbol{w}}^{(T)}\right)-F\left(\boldsymbol{w}^*\right) \leq \frac{1}{T\left(\omega\eta\left(1-\frac{\beta\eta}{2}\right)-\frac{\rho\left(\phi(\tau)+\frac{p_U}{M}\left[(1-\alpha)M\Theta+\frac{2\sqrt{M}\sigma}{\pi}\right]\right)}{\tau\varepsilon^2}\right)}. \quad (24)$$

This completes the proof.

APPENDIX B
PROOF OF PROPOSITION 3

First, we define $H(\tau)$ as

$$H(\tau)\triangleq\frac{\phi(\tau)+\nabla}{\tau} \quad (25)$$

where $\phi(\tau)=(\delta/\beta)((\eta\beta+1)^\tau-1)-\eta\delta\tau$ and $\nabla=(p_U/M)[(1-\alpha)M\Theta+[(2\sqrt{M}\sigma)/\pi]]>0$, respectively.

With a slight abuse of τ, we consider continuous values of $\tau>1$, and then have

$$H'(\tau)=-\frac{\nabla}{\tau^2}+\frac{\delta(\eta\beta+1)^\tau(\ln(\eta\beta+1)^\tau-1)+\delta}{\beta\tau^2} \quad (26)$$

and

$$\begin{aligned}H''(\tau)&=\frac{\delta}{\beta\tau^4}\Big(\tau^3(\eta\beta+1)^\tau\ln^2(\eta\beta+1)\\&\quad-2\tau(\eta\beta+1)^\tau\left(\ln(\eta\beta+1)^\tau-1\right)-2\tau\Big)+\frac{2\nabla}{\tau^3}\\&=\frac{\delta}{\beta\tau^3}\Big((\eta\beta+1)^\tau\Big(\left(\ln(\eta\beta+1)^\tau-1\right)^2+1\Big)-2\Big)+\frac{2\nabla}{\tau^3}. \quad (27)\end{aligned}$$

We then define $f(x)$ as

$$f(x)\triangleq x\Big(\left(\ln x-1\right)^2+1\Big)-2 \quad (28)$$

then, the derivative of $f(x)$ can be expressed as

$$\begin{aligned}f'(x)&=(\ln x-1)^2+2(\ln x-1)+1\\&=(\ln x-1+1)^2\\&=\ln^2 x\geq 0. \quad (29)\end{aligned}$$

Note that $(\eta\beta+1)^\tau\geq 1$ since $\tau\geq 1$, $\eta\beta\geq 0$, we can know that

$$f(\eta\beta+1)^\tau\geq f(1)=0. \quad (30)$$

Combining (27), (28), and (30), we have

$$H''(\tau)\geq 0. \quad (31)$$

Therefore, under the unreliable behaviors of clients with a fixed T, the convergence upper bound is a convex function of the number of local epochs τ. This completes the proof.

REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] J. Li, S. Chu, F. Shu, J. Wu, and D. N. K. Jayakody, "Contract-based small-cell caching for data disseminations in ultra-dense cellular networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1042–1053, May 2019.

[3] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preserving location data publishing: A machine learning approach," *IEEE Trans. Knowl. Data Eng.*, early access, Jan. 7, 2020, doi: 10.1109/TKDE.2020.2964658.

[4] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preservation in location-based services: A novel metric and attack model," *IEEE Trans. Mobile Comput.*, early access, May 11, 2020, doi: 10.1109/TMC.2020.2993599.

[5] D. C. Nguyen *et al.*, "Enabling AI in future wireless networks: A data life cycle perspective," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 553–595, 1st Quart., 2021.

[6] P. Vepakomma, T. Swedish, R. Raskar, O. Gupta, and A. Dubey, "No peek: A survey of private distributed deep learning," 2018. [Online]. Available: arXiv:1812.03288.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 54. Fort Lauderdale, FL, USA, Apr. 2017, pp. 1273–1282.

[8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[9] W. Yu *et al.*, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2017.

[10] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.

[11] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[12] Y. Peng, A. Tan, J. Wu, and Y. Bi, "Hierarchical edge computing: A novel multi-source multi-dimensional data anomaly detection scheme for industrial Internet of Things," *IEEE Access*, vol. 7, pp. 111257–111270, 2019.

[13] H. Peng, S. Si, M. K. Awad, N. Zhang, H. Zhao, and X. S. Shen, "Toward energy-efficient and robust large-scale WSNs: A scale-free network approach," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 4035–4047, Dec. 2016.

[14] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Proc. Adv. Neural Inf. Process. Syst. 32: Ann. Conf. Neural Inf. Process. Syst. (NeurIPS)*, Vancouver, BC, Canada, Dec. 2019, pp. 8632–8642. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/ec1c59141046cd1866bbbcdfb6ae31d4-Abstract.html

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. 23rd Int. Conf. Artif. Intell. Stat. (AISTATS)*, vol. 108. Palermo, Italy, Aug. 2020, pp. 2938–2948.

[16] L. Zhao *et al.*, "Shielding collaborative learning: Mitigating poisoning attacks through client-side detection," *IEEE Trans. Dependable Secure Comput.*, early access, Apr. 14. 2020, doi: 10.1109/TDSC.2020.2986205.

[17] S. Fu, C. Xie, B. Li, and Q. Chen, "Attack-resistant federated learning with residual-based reweighting," 2019. [Online]. Available: http://arxiv.org/abs/1912.11464.

[18] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2020.
[19] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "PDGAN: A novel poisoning defense method in federated learning using generative adversarial network," in *Proc. Algorithms Archit. Parallel Process.*, Cham, Switzerland, 2020, pp. 595–609.
[20] L. Muñoz-González, K. T. Co, and E. C. Lupu, "Byzantine-robust federated machine learning through adaptive model averaging," Sep. 2019. [Online]. Available: https://arxiv.org/abs/1909.05125.
[21] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Security Symp. (USENIX Security)*, Aug. 2020, pp. 1605–1622.
[22] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018. [Online]. Available: arXiv:1806.00582.
[23] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Stockholm, Sweden, Jul. 2018, pp. 3219–3227.
[24] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.
[25] S. U. Stich, "Local SGD converges fast and communicates little," 2018. [Online]. Available: arXiv:1805.09767.
[26] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 119–129.
[27] E. D. B. Solis, A. M. Neto, and B. N. Huallpa, "Real-time collision risk estimation based on pearson's correlation coefficient: Comparative analysis with real distance from the Velodyne 3D laser scanner," in *Proc. 13th Latin Amer. Robot. Symp. 4th Brazil. Robot. Symp. (LARS/SBR)*, Recife, Brazil, 2016, pp. 234–238.
[28] Y. Liu *et al.*, "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6348–6358, Apr. 2021.
[29] T. Luo and S. G. Nagarajan, "Distributed anomaly detection using autoencoder neural networks in WSN for IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, 2018, pp. 1–6.
[30] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 1467–1474.
[31] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, Apr. 2014.
[32] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 634–643.
[33] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 603–618.
[34] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Paris, France, 2019, pp. 2512–2520.
[35] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy (SP)*, San Francisco, CA, USA, 2019, pp. 691–706.
[36] L. Zhao, Q. Wang, Q. Zou, Y. Zhang, and Y. Chen, "Privacy-preserving collaborative deep learning with unreliable participants," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1486–1500, 2019.

Chuan Ma (Member, IEEE) received the B.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and the Ph.D. degree from the University of Sydney, Sydney, NSW, Australia, in 2018.

He is currently working as a Lecturer with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He has published more than ten journal and conference papers, including a best paper in WCNC 2018. His research interests include stochastic geometry, wireless caching networks, and machine learning, and currently focuses on the big data analysis and privacy preservation.

Jun Li (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009.

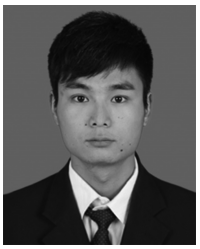
From January 2009 to June 2009, he worked with the Department of Research and Innovation, Alcatel Lucent Shanghai Bell, Shanghai, as a Research Scientist. From June 2009 to April 2012, he was a Postdoctoral Fellow with the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW, Australia. From April 2012 to June 2015, he was a Research Fellow with the School of Electrical Engineering, University of Sydney, Sydney. Since June 2015, he has been a Professor with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China. He was a Visiting Professor with Princeton University, Princeton, NJ, USA, from 2018 to 2019. His research interests include network information theory, game theory, distributed intelligence, multiple agent reinforcement learning, and their applications in ultradense wireless networks, mobile-edge computing, network privacy and security, and Industrial Internet of Things. He has coauthored more than 200 papers in IEEE journals and conferences, and holds one U.S. patent and more than ten Chinese patents in these areas.

Dr. Li received the Exemplary Reviewer of IEEE TRANSACTIONS ON COMMUNICATIONS in 2018, and the Best Paper Award from IEEE International Conference on 5G for Future Wireless Networks in 2017. He was serving as an Editor for IEEE COMMUNICATION LETTERS and a TPC member for several flagship IEEE conferences.

Ming Ding (Senior Member, IEEE) received the B.S. and M.S. degrees (with First-Class Hons.) in electronics engineering and the Ph.D. degree in signal and information processing from Shanghai Jiao Tong University, Shanghai, China, in 2004, 2007, and 2011, respectively.

From April 2007 to September 2014, he worked with the Sharp Laboratories of China, Shanghai, as a Researcher/Senior Researcher/Principal Researcher. He is currently a Senior Research Scientist with Data61, CSIRO, Sydney, NSW, Australia. He has authored over 140 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as a book *Multi-Point Cooperative Communication Systems: Theory and Applications* (Springer). He also holds 21 U.S. patents and co-invented another 100+ patents on 4G/5G technologies in CN, JP, KR, and EU. His research interests include information technology, data privacy and security, machine learning, and AI.

Dr. Ding is currently an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE WIRELESS COMMUNICATIONS LETTERS. Besides, he has served as a guest editor/co-chair/co-tutor/TPC member for many IEEE top-tier journals/conferences and received several awards for his research work and professional services.



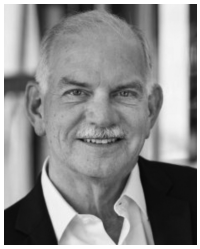
Kang Wei (Graduate Student Member, IEEE) received the B.Sc. degree in information engineering from Xidian University, Xi'an, China, in 2014, and the M.Sc. degree from the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing, China, in August 2018, where he is currently pursuing the Ph.D. degree.

His current research interests include data privacy and security, differential privacy, AI and machine learning, information theory, and channel coding theory in NAND flash memory.

Wen Chen (Senior Member, IEEE) received the B.S. and M.S. degrees from Wuhan University, Wuhan, China, in 1990 and 1993, respectively, and the Ph.D. degree from the University of Electro-Communications, Tokyo, Japan, in 1999.

He is a tenured Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is the Director of Broadband Access Network Laboratory. He has published more than 100 papers in IEEE journals and more than 100 papers in IEEE conferences, with citations more than 6000 in Google Scholar. His research interests include multiple access, wireless AI, and metasurface communications.

Dr. Chen is the Shanghai Chapter Chair of the IEEE Vehicular Technology Society and an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE ACCESS, and IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY. He is a Fellow of the Chinese Institute of Electronics and the Distinguished Lecturer of the IEEE Communications Society and IEEE Vehicular Technology Society.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University, Princeton, NJ, USA, in 1977.

From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana–Champaign, Urbana, IL, USA. Since 1990, he has been on the faculty at Princeton, where he is the Michael Henry Strater University Professor. From 2006 until 2016, he also served as the Dean of Princeton's School of Engineering and Applied Science. His research interests are in the areas of information theory, machine learning, and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications* (Cambridge University Press, 2021).

Dr. Poor is a member of the National Academy of Engineering and the National Academy of Sciences, and is a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo, awarded in 2019.