

Byzantine-Robust Aggregation in Federated Learning Empowered Industrial IoT

Shenghui Li, *Student Member, IEEE*, Edith Ngai, *Senior Member, IEEE*, and Thiemo Voigt, *Member, IEEE*

Abstract—Federated Learning (FL) is a promising paradigm to empower on-device intelligence in Industrial Internet of Things (IIoT) due to its capability of training machine learning model across multiple IIoT devices, while preserving the privacy of their local data. However, this distributed architecture of FL relies on aggregating the parameter list from the remote devices, which poses potential security risks caused by malicious devices. In this paper, we propose a flexible and robust aggregation rule, called Auto-weighted Geometric Median (AutoGM), and analyze the robustness against outliers in the inputs. To obtain the value of AutoGM, we design an algorithm based on alternating optimization strategy. Using AutoGM as aggregation rule, we propose two robust FL solutions AutoGM_FL and AutoGM_PFL. AutoGM_FL learns a shared global model using the standard FL paradigm, and AutoGM_PFL learns a personalized model for each device. We conduct extensive experiments on the FEMNIST and Bosch IIoT datasets. The experimental results show that our solutions are robust against both model poisoning and data poisoning attacks. In particular, our solutions sustains high performance even when 30% of the nodes perform model or 50% of the nodes data poisoning attacks.

Index Terms—Federated learning (FL), industrial Internet of Things (IIoT), Byzantine-robust, geometric median, security and privacy.

I. INTRODUCTION

MACHINE learning techniques are increasingly used in Industrial Internet of Things (IIoT) applications, such as smart grid, autonomous driving, and smart factory [1–3]. Nevertheless, the collection of user data for centralized training causes serious privacy concerns, i.e., some users want to keep their local data private and may not want to transmit their local data to the centralized server [4]. In addition, the massive amount of data generated in IIoT systems poses a challenge on communication efficiency between the centralized server and the remote IoT devices [5], which also limits the scalability of the system. Due to the rapid developments in computing hardware, IoT devices are nowadays equipped with more powerful computation and storage capabilities. As a consequence, machine learning models can be deployed on IoT devices to support on-device training at the network edge [6].

Federated Learning (FL) is a promising paradigm to empower on-device intelligence and mitigate the privacy and scalability

issues in IIoT systems [7, 8]. FL allows massive participants to learn the statistical models collaboratively by sharing model parameters while keeping their data in local devices for privacy preservation. Different from the gradient aggregation techniques in traditional distributed machine learning, typical FL approaches only select a subset of devices at each round and aggregate the updated parameters after several local steps of stochastic gradient descent (SGD), which dramatically reduces the communication overhead in large scale IIoT network systems [9]. Benefiting from these advantages, FL has recently been applied to solve data-driven problems in IIoT applications, such as anomaly detection [10] as well as data and resource management [11].

One of the major security threats to the model training process of FL are Byzantine failures, where participants (known as Byzantine nodes) may upload arbitrary parameters to the central server in order to degrade the performance of the global model [12, 13]. In Federated Learning empowered IIoT (FL-IIoT) systems, Byzantine failure results from poisoning attacks [10], abnormal sensors [14], communication noise [15], etc. Among them, poisoning attacks are particularly challenging as the hackers can deliberately manipulate the training process either by directly modifying local model parameters sent from the local devices to the coordination server (a.k.a. model poisoning attack), or by changing the samples in the local training set to bias the local model (a.k.a. data poisoning attack) [12], which could pose a serious threat to the security of the learning system. More importantly, it has been shown that even a single malicious participant can significantly alter the trained model [16]. Therefore, it is imperative to make FL robust to Byzantine failures, in particular for security-critical IIoT applications such as automated guided vehicles and robot motion control [17].

In terms of defense, Geometric Median (GM) has been widely used for gradient aggregation in the literature of robust distributed learning [18–20] because it can tolerate up to half malicious devices and estimate the underlying true parameters. In addition, GM shows convergence properties when it acts as an aggregation rule in gradient descent methods [21]. Recently, it has also been applied to aggregate parameters in robust FL solutions [12, 21]. However, when the local datasets are imbalanced, i.e., the number of samples varies among devices, their solution lacks guarantees in terms of robustness. Moreover, our experimental results show that existing GM-based solutions [21] cannot cope well with model poisoning attacks. Therefore, it is important to enhance the robustness of GM.

In addition, due to the heterogeneity of the data in IIoT systems, a single robust global model may only perform well

This research was partly supported by the RGC Grant Research Fund no. 17203320 from Hong Kong. (Corresponding author: Edith Ngai.)

Shenghui Li is with the Department of Information Technology, Uppsala University, Uppsala, Sweden (e-mail: shenghui.li@it.uu.se).

Edith Ngai is with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong, China (e-mail: chngai@eee.hku.hk).

Thiemo Voigt is with the Department of Information Technology, Uppsala University, Uppsala, Sweden, and also with RISE, the Research Institutes of Sweden, Stockholm, Sweden (e-mail: thiemo.voigt@it.uu.se).

in some of the devices [22]. To tackle the heterogeneity challenge, one effective way is to perform Personalized Federated Learning (PFL) in the system, e.g., in addition to the shared global model, each device trains a personalized model using its local data and under the guidance of the global model [23]. However, very few studies have investigated the robustness of PFL [24], especially in the context of Byzantine failures.

In this paper, we study a Byzantine-robust FL problem in IIoT systems, and aim to mitigate the challenges of model poisoning and data poisoning attacks during model training process. We propose *AutoGM*, a secure aggregation rule that enhances the robustness of GM in order to learn a robust global model. By applying *AutoGM* as an aggregation rule for traditional FL and PFL paradigms, we propose *AutoGM_FL*, and *AutoGM_PFL*, respectively. *AutoGM_FL* learns a shared global model using the standard FL paradigm, whereas *AutoGM_PFL* learns a personalized model for each device to further address the data heterogeneity challenge. We summarize our contributions below:

- We propose Auto-weighted Geometric Median (*AutoGM*), a flexible robust aggregation rule for a given set of parameter vectors. We show that *AutoGM* achieves robustness by automatically excluding the extreme outliers and re-weighting the rest of the points according to a user-specified threshold of skewness. We design an algorithm based on alternating optimization strategy to compute the position of *AutoGM*.
- We integrate *AutoGM* with both standard FL and PFL paradigms for FL-IIoT system, and propose robust solutions *AutoGM_FL* and *AutoGM_PFL*, respectively, both of which show robustness against poisoning attacks, and inherit the privacy and scalability properties of traditional FL.
- We conduct extensive experiments to evaluate the proposed solutions using two real-world datasets. We successfully demonstrate the superior robustness of our approach against model poisoning attacks and data poisoning attacks compared to the state-of-the-art approaches. In particular, our solutions sustains high performance even when 30% of the nodes perform model or 50% of the nodes data poisoning attacks.

II. RELATED WORK

A. Federated Learning on Industrial IoT

Recent years have witnessed an increasing research interest in FL within the context of IIoT as it preserves user privacy by enabling local model training at the edge. Many works concentrate on FL-based anomaly detection in IIoT [10, 25–27]. For example, Taheri et al. [10] presented a robust FL based architecture called Fed-IIoT for detecting Android malware applications in IIoT. On the participant side, data are triggered by two dynamic poisoning attacks based on a generative adversarial network (GAN) and federated GAN. The server side of Fed-IIoT monitors the global model and shapes a robust collaboration training model, by avoiding anomalies in aggregation by a GAN network (A3GAN) and by adjusting

two GAN-based countermeasure algorithms. Another line of work focuses on data and resource management [11, 28, 29]. For instance, Zhang et al. [28] presented an FL framework to manage IIoT equipment data in wireless network environments. In order to increase the model aggregation rate and reduce communication costs, the authors applied deep reinforcement learning (DRL) to the IIoT device selection process, specifically to select those devices with accurate models. Furthermore, the authors proposed an FL algorithm assisted by DRL, which takes into account the privacy and efficiency of data training of the IIoT system. In addition, there were some typical representatives who applied FL to other interesting IIoT applications such as digital twins [30], smart cities [31], and healthcare systems [32].

B. Robust Federated Learning

Training time attack is a critical threat to the security of the FL learning systems due to decentralization of data and the requirement of collaborative training from multiple devices [12, 33, 34]. Poisoning attacks in FL can be classified into two categories, data poisoning attacks and model poisoning attacks. Data poisoning attacks aim to modify the training datasets in local devices, while model poisoning attacks directly manipulate the local models sent from the local devices to the coordination server [12]. For example, Bagdasaryan et al. [33] introduced the model replacement attack to backdoor FL models and concluded that FL is generically vulnerable to model poisoning attacks. Konstantinov and Lampert [35] showed that datasets from the edge devices vary greatly in quality, reliability and relevance for the learning task.

Robust aggregation has been explored to handle the devices sending corrupted updates to the server, including geometric median (GM) [18], Krum [16], trimmed mean [20], and median [20]. They are commonly used to estimate the model parameters and mitigate the effect of malicious updates in global aggregation. The coordination server may follow different aggregation rules to generate a robust global model from the local updates. Among those aggregation rules, GM is particularly popular in robust statistics, given that it can defend against up to 50% outliers in the parameter list [36]. In 2017, Chen et al. [18] proposed a GM-based method to aggregate the gradients for distributed statistical machine learning and showed the robustness and convergence in i.i.d settings. Recently, Pillutla et al. [21] applied GM as a robust aggregator for FL and analyze the convergence of the resulting FL algorithm for i.i.d. least-squares objective. However, for imbalanced datasets where the local dataset size are different among devices, their solution lacks theoretical guarantees in terms of robustness. Compared to GM, our proposed aggregation rule is more flexible as it can be tuned based on the desired amount of skewness; GM can in fact be seen as a special case of *AutoGM*.

III. FEDERATED LEARNING FOR IIoT SYSTEMS

In this section, we introduce the system model and threat models of the Federated Learning empowered IIoT (FL-IIoT) system considered in this work. An overview is illustrated in Fig. 1.

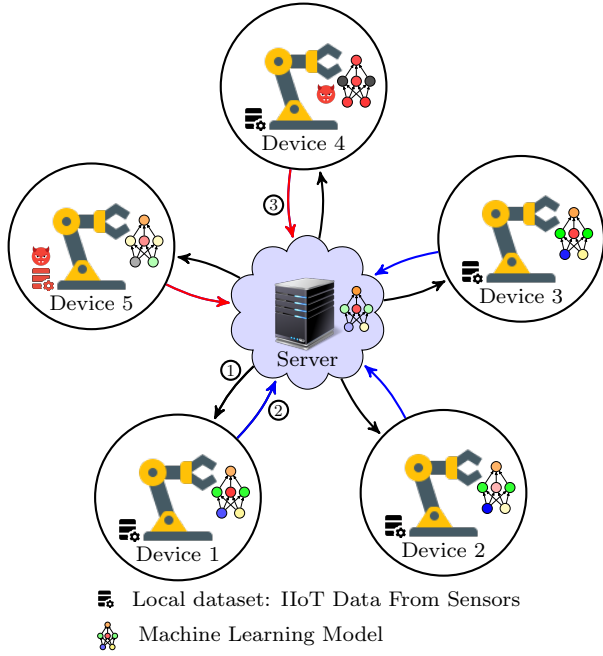


Fig. 1: Overview of the FL-IIoT System, where multiple IIoT devices collaboratively train a machine learning model using their local datasets. The central server coordinates the training by aggregating local model and broadcasting the resulting model to the devices in each training round. Some malicious devices could send corrupted model (as shown in ③) due to data poisoning attack (e.g., Device 4), or model poisoning attack (e.g., Device 5).

A. System Model

We consider an IIoT system that involves multiple IIoT devices and one central server for collaborative model training. More details are described as follows.

IIoT Device: Each device owns a private dataset and the replica of a shared global model. During the each round of training process, a subset of devices are selected for model updates. Each selected device then receives the latest global model from the central server (as denoted by ① in Fig. 1) and updates the model with several steps of stochastic gradient descent (SGD) using its private dataset, after which it uploads the new parameters to the server via wireless network (as denoted by ② in Fig. 1).

Server: The server is responsible for building a comprehensive robust global machine learning model by aggregating the model parameters from the locally learned models uploaded from the IIoT devices. After aggregating the local parameters, it broadcasts the resulting global model to the devices to enable the next rounds of local updating. Multiple rounds of interactions between the server and devices are required to obtain an accurate shared global model.

In IIoT systems, wireless networks are widely used for the communication between the server and the IIoT devices because of their flexibility and portability. Considering the limited capability of wireless networks, only a random subset of the devices is chosen in each round of training to avoid network

congestion, which allows a large number of participants for collaborative training.

B. Threat Models

In this paper, we aim to address the threat of poisoning attacks in FL-IIoT systems, where the devices can upload poisoned local model parameters to the server (as denoted by ③ in Fig. 1). In this context, poisoning attacks are a major concern since they may cause Byzantine failures. Poisoning attacks in FL-IIoT mainly covers model poisoning attacks and data poisoning attacks [37]. For model poisoning attacks, malicious devices craft the local models by directly changing the parameters or adding noise [12], so that the model parameters from local training are manipulated (e.g., Device 4 in Fig. 1). For data poisoning attacks, corrupted devices are honest in conducting local model training but their training datasets are poisoned with flipped or noisy labels (e.g., Device 5 in Fig. 1) [23].

Like many other studies on poisoning attacks [12, 23, 38], we assume that the goal of the attackers is to manipulate the trained global model and make it unusable for the system, i.e., cause a high error rate indiscriminately for testing data. Such attacks are known as untargeted poisoning attacks [12]. Although we particularly focus on model poisoning and data poisoning attacks in this paper, we remark that the solutions we propose are Byzantine-robust in general, and have the potential to handle other types of Byzantine failures, as our solutions enhance the robustness of the classic Geometric Median (GM), which is proven as a Byzantine-robust aggregation rule [18].

We make the following assumptions in this paper [12, 21, 39]: (1) Each adversary controls exactly one non-colluding and corrupted device. Since devices do not collude, the effect of malicious updates from each adversary to the global model is limited [39]. (2) No more than half of the devices are malicious, so that the majority of the devices can provide benign model parameters which can be aggregated to a consensus global model [12]. (3) The communication channels are reliable, which means that we do not consider communication noise and failures. As we majorly focus on model poisoning attack and data poisoning attack, communication issues will further complicate the problem.

IV. AUTO-WEIGHTED GEOMETRIC MEDIAN

Before presenting our robust federated learning algorithm in FL-IIoT, we first define a general robust aggregation rule for a given set of vectors, named Auto-weighted Geometric Median (AutoGM). AutoGM achieves robustness by reweighting the inputs according to their Euclidean distances. We study the robustness of AutoGM and design an algorithm to solve it.

A. GM and AutoGM

The robust aggregation rule we propose in this section is generalized from Geometric Median (GM), which is defined as:

Definition 1 (Geometric Median (GM)). The geometric median of a collection of vectors $S = \{\mathbf{z}_k\}_{k \in [K]}$ is a point $\hat{\mathbf{z}}$ that

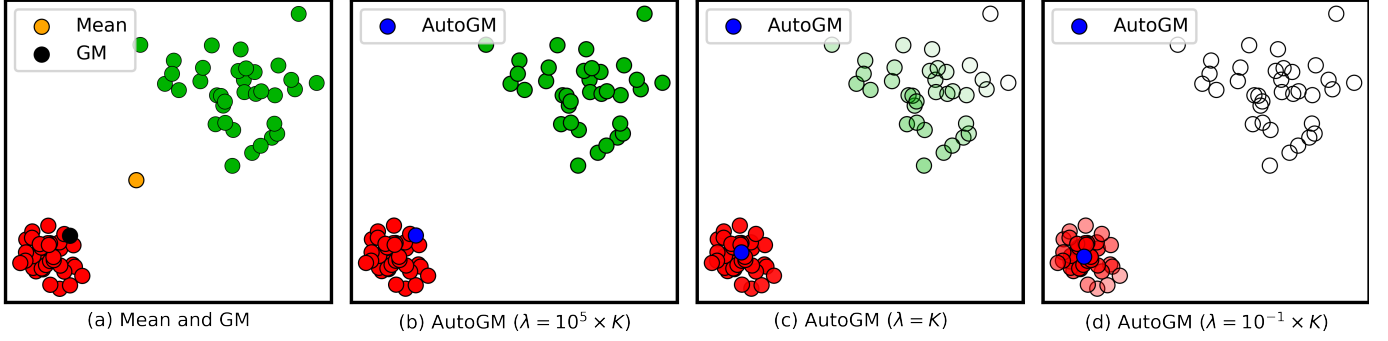


Fig. 2: Illustration of Mean, GM, and AutoGM of benign points (in red) with outliers (in green). (a) shows that the Mean is driven away by outliers, and GM lies on the edge of benign points. (b)-(d) show the positions of AutoGM with different λ . The transparency of each data point in (b)-(d) reflects its weight to AutoGM, and the white points are zero-weighted.

minimizes the sum of its Euclidean distances to all the points in S :

$$\hat{z} := \operatorname{argmin}_z \sum_{k \in [K]} \|z - z_k\|, \quad (1)$$

where $K = |S|$, and $\|\cdot\| = \|\cdot\|_2$ is the Euclidean norm.

GM is a point representative of a given set of points that is resistant to outliers. Specifically, GM is stable even if we arbitrary corrupt up to half of the inputs.

Cohen et al. have shown [40] that as long as the majority of the points are within the Euclidean ball centered at z , then GM must lie in the Euclidean ball with the same center z while blown up by a constant factor only. GM has typically been applied as a gradient aggregation rule in the literature of Byzantine-robust distributed learning and shows convergence properties [18–20].

However, the robustness of GM is limited as the constant factor is still impacted by the number of outliers [18], i.e., the bound increases as the size of outliers rises. Based on GM, we further propose a more flexible robust aggregation rule AutoGM, which is defined as:

Definition 2 (Auto-weighted Geometric Median (AutoGM)). We denote the Auto-weighted Geometric Median of $S = \{z_k\}_{k \in [K]}$ by $\tilde{z}^* = \text{AutoGM}(S)$. Together with the weight vector α of S , we can obtain their values by solving the following problem:

$$\begin{aligned} \tilde{z}^*, \tilde{\alpha}^* := \operatorname{argmin}_{z, \alpha} \sum_{k \in [K]} \alpha_k \|z - z_k\| + \frac{\lambda}{2} \sum_{k \in [K]} \alpha_k^2, \quad (2) \\ \text{s.t. } \alpha \in \mathbb{R}_+^K, \mathbf{1}^\top \alpha = 1, \end{aligned}$$

where λ is a user-specified hyper-parameter that controls the smoothness of α .

An intuitive interpretation of the above objective is that the first term aims to find the weighted GM of S and allocates higher weights to the points closer to z , and allocates lower weights to the points that are far from z . In contrast, the second term encourages the smoothness of the weights. Specifically, the second term is minimized when the weights are distributed equally. Thus, it acts as a form of regularization, by encouraging the usage of information from as many points as possible.

B. Analysis of AutoGM

We show that AutoGM achieves robustness by automatically excluding the extreme outliers and re-weighting the rest of the points according to a user-specified threshold of skewness. To illustrate this, we first present two definitions.

Definition 3 (L_1 -skewness). For a given point z and a collection of vectors $S = \{z_k\}_{k \in [K]}$, we define L_1 -skewness as:

$$R(z, S) := \max_{k \in [K]} (\|z - z_k\|) - \frac{1}{K} \sum_{k \in [K]} \|z - z_k\|. \quad (3)$$

Here we use the symbol “ L_1 ” because the last term is equivalent to the L_1 norm of Euclidean distance divided by K . To illustrate this, consider a vector $d \in \mathbb{R}^K$, in which coordinates are given by the distances: $d_k = \|z - z_k\|$. Then the formula becomes:

$$R(z, S) := \max_{k \in [K]} (d_k) - \frac{1}{K} \|d\|_1. \quad (4)$$

where $\|\cdot\|_1$ is the L_1 norm. In other words, L_1 -skewness is also defined as the difference between the maximum and the average Euclidean distances to the point z .

Definition 4 (S_p of AutoGM). We define S_p as the largest subset of S in which the solutions of weights for AutoGM are non-zero, i.e. $\tilde{\alpha}_k^* \neq 0, \forall k \in S_p$ and $\tilde{\alpha}_k^* = 0, \forall k \in S \setminus S_p$.

Based on the definitions above, we show in Theorem 1 that for any given z , there is a closed-form solution that minimizes the subproblem in Equation (2) with respect to α . A proof of Theorem 1 is provided in Appendix A.

Theorem 1. Given any z in Equation (2), S_p will satisfy the following condition:

$$R(z, S_p) \leq \frac{\lambda}{m}, \quad (5)$$

where $m = |S_p|$ is the size of S_p .

Then, the optimal weights $\tilde{\alpha}^*$ are given by

$$\tilde{\alpha}_k^* = \begin{cases} \frac{1}{m} + \frac{1}{\lambda} \left(\frac{1}{m} \sum_{k \in S_p} \|z - z_k\| - \|z - z_k\| \right) & \text{if } k \in S_p, \\ 0 & \text{if } k \notin S_p. \end{cases} \quad (6)$$

The theorem above will help us to solve the joint optimization problem in Equation (2) as we will show in Section IV-C. Now we study the relation between α and z when the solution is optimal. Given $z = \tilde{z}^*$ as the optimal AutoGM point in Theorem 1, we have the following observations:

Limited L_1 -skewness: $\frac{\lambda}{m}$ limits the L_1 -skewness $R(\tilde{z}^*, S_p)$ by excluding the points that are far from $\tilde{z}^*$. Note that S_p satisfies $R(\tilde{z}^*, S_p) \leq \frac{\lambda}{m}$, that is, the difference between maximum and average distances are limited by $\frac{\lambda}{m}$ (Equation (3)). The points that are significantly distant to $\tilde{z}^*$ are excluded by S_p and thus zero-weighted by the objective. This property protects $\tilde{z}^*$ against being skewed by extreme outliers.

Re-scaled weights: $\tilde{\alpha}^*$ re-scales the weight of each point in S_p according to Euclidean distance. According to Equation (6), when $z = \tilde{z}^*$, the points that are closer to $\tilde{z}^*$ gain higher weights as the term $\|z - z_k\|$ is smaller. That is to say, the distant points are considered to be less important to determine the position of $\tilde{z}^*$. This property further reduces the impact of outliers to AutoGM.

The hyperparameter λ controls the trade-off between exploiting fewer points and leveraging information from all input points. As $\lambda \rightarrow \infty$, the L_1 -skewness is unlimited, which means all points are included in S_p and assigned with the same weight $\frac{1}{m}$, regardless of their distances to $\tilde{z}^*$. Then the problem is reduced to the naive GM as in Definition (1). In contrast, as $\lambda \rightarrow 0$, the weights become extremely sparse as the objective encourages $\tilde{z}^*$ to be close to (or even overlap with) its nearest points. It is noticeable that λ also controls the re-scaling extent of each point in S_p (the second term in Equation (6)). In practice, the optimal λ is unknown as the true L_1 -skewness of the benign point is usually unknown. In this paper, we empirically set $\lambda = K$ (where $K = |S|$) by default.

As an example, Fig. 2 demonstrates the impact of outliers on the naive Mean, GM, and AutoGM. There are 40 benign points gathering in the bottom left corner of the space and 35 outliers scattering in the opposite corner. In Fig. 2-(a), the Mean is obviously driven away by the outliers. GM lies on the edge of benign points, which is less influenced by the outliers. Fig. 2-(b) to Fig. 2-(d) show the positions of AutoGM with different λ . When $\lambda = 10^5 \times K$, AutoGM lies at the same point as naive GM (Fig. 2-(b)). When $\lambda = K$, the outliers are down-weighted and the position of AutoGM is more central to the benign points (Fig. 2-(c)), which mainly benefits from the re-scaling of the weights. When $\lambda = 10^{-1} \times K$, AutoGM locates at a similar position as that of Fig. 2-(c), and all the outliers are zero-weighted (Fig. 2-(d)). We conclude that AutoGM enhances the robustness of GM when λ is small.

C. Optimizing AutoGM

We notice that α and w in Equation (2) are dependent on each other, which makes the problem hard to solve. However, when either α or w is fixed, the problem can be solved easily. When w is fixed, thanks to the existence of a closed-form solution (Theorem 1), the optimal weights $\tilde{\alpha}^*$ can be computed efficiently. Furthermore, by fixing $\alpha = \tilde{\alpha}^*$, the problem becomes finding the naive weighted GM

$$\tilde{z} := \underset{z}{\operatorname{argmin}} \sum_{k \in S} \tilde{\alpha}_k^* \|z - z_k\|. \quad (7)$$

Algorithm 1 Optimization of AutoGM

```

1: Input:  $K, \{z_k\}_{k \in S}, T_g$ 
2:  $\tilde{\alpha}_k \leftarrow 1/K, \forall k \in S$ 
3: repeat
4:   Compute  $\tilde{z}$  by optimizing Problem (7)
5:    $\tilde{\alpha} \leftarrow \text{Reweight}(\{\|\tilde{z} - z_k\|\}_{k \in S})$ 
6: until convergence
7: return  $\tilde{z}, \tilde{\alpha}$ 

```

Subroutine Reweight

```

8: Input:  $\{\|z - z_k\|\}_{k \in S}$ 
9: Sort  $\{\|z - z_k\|\}_{k \in S}$  in increasing order
10:  $\eta \leftarrow 0, \gamma \leftarrow 0$ 
11: for  $p = 1, \dots, K$  do
12:    $\gamma \leftarrow \gamma + \|z - z_p\|$ 
13:    $\tilde{\eta} \leftarrow (\gamma + \lambda)/p$ 
14:   if  $\tilde{\eta} < \|z - z_p\|$  then
15:     Break
16:   else
17:      $\eta \leftarrow \tilde{\eta}$ 
18:   end if
19: end for
20:  $\tilde{\alpha}_k^* \leftarrow (\eta - \|z - z_k\|)/\lambda, \forall k < p$ 
21:  $\tilde{\alpha}_k^* \leftarrow 0, \forall k \geq p$ 
22: return  $\tilde{\alpha}^*$ 

```

Based on the observations above, we propose an alternating optimization approach to solve AutoGM. The key idea is to divide the problem into two parts, i.e., one sub-problem for estimating the weighted GM, $\tilde{z}$, and the other sub-problem for automatically weighting the importance of each point. Then we minimize the objective iteratively w.r.t. one variable each time while fixing the other one. In this way, we compute the AutoGM as shown in Algorithm 1, which consists of the following two alternating steps:

Updating z . When α is fixed, we solve Problem (7) to obtain the weighted GM. Although there is no closed-form solution to the GM problem, a $(1 + \epsilon)$ -approximate solution can be computed in nearly linear time [40]. In this paper, we solve the problem using the smoothed Weiszfeld algorithm proposed by Pillutla et al. [21].

Updating α . When z is fixed, we update α according to Theorem 1 as shown in the Subroutine of Algorithm 1. To provide a clearer explanation of the idea, we rewrite Equation (6) as

$$\tilde{\alpha}_k^* = \frac{1}{\lambda} [\eta - \|z - z_k\|]_+, \quad (8)$$

where

$$\eta = \frac{1}{m} \left(\lambda + \sum_{k \in S_p} \|z - z_k\| \right), \quad (9)$$

and $[\cdot]_+ = \max(0, \cdot)$. For a given z , we firstly sort the points in S according to their Euclidean distance to z (Line 9). Then we can obtain η by a linear search and checking the satisfaction in Equation (5) (Line 11-19). Finally we reallocate the weights according to Equation (8) (Line 21-22).

According to Cohen et al. [40], a $(1 + \epsilon)$ -approximate weighted GM of K points in $\mathbb{R}^d$ can be computed in $O(Kd \log^3 \epsilon^{-1})$ time. The routine of updating α only takes $O(K)$ time. Thus, the total time complexity of AutoGM is $O(T_g K d \log^3 \epsilon^{-1})$, where T_g is the number of iteration.

V. ROBUST FEDERATED LEARNING

In this section, we first formulate the FL problem for the IIoT system. Then we propose a robust solution AutoGM_FL to train a global model using AutoGM to aggregate the parameters. Finally, we propose the extension to robust PFL, named AutoGM_PFL, to simultaneously learn a shared robust global model and a personalized local model for each device.

A. Problem Formulation

Recall that in the FL-IIoT system we introduced in Section III, several devices collaboratively learn a shared global model assisted by the coordination of a central server. The goal is to find a parameter vector $\mathbf{w}$ that minimizes the following distributed optimization model:

$$\min_{\mathbf{w}} F(\mathbf{w}) = \min_{\mathbf{w}} \sum_{k \in [N]} p_k F_k(\mathbf{w}), \quad (10)$$

where N is the total number of devices and p_k is the weight of the k -th device. Let n_k be the number of samples available in the k -th device, we have $p_k = \frac{n_k}{\sum_{k \in [N]} n_k}$. The local objective $F_k(\cdot)$ can be defined by empirical risks over local data, i.e., $F_k(\mathbf{w}) = \frac{1}{n_k} \sum_{j \in [n_k]} \ell(\mathbf{w}; x_{k,j})$, where $\ell(\cdot; \cdot)$ is a user-specified loss function and $x_{k,j}$ is a training sample.

If we let $\tilde{F}_k(\mathbf{w}) = p_k N F_k(\mathbf{w})$, then the global objective becomes a simple average of all scaled local objectives:

$$F(\mathbf{w}) = \frac{1}{N} \sum_{k \in [N]} \tilde{F}_k(\mathbf{w}). \quad (11)$$

The objective above is aligned with the formulation in [41], which simplifies the analysis of convergence properties.

B. Robust Federated Learning with AutoGM

The optimization of the global objective above is similar to the case of FedAvg [7, 41], where a subset of the total devices are selected at each global round and run SGD locally for E_g iterations, and then the resulting model parameters are uploaded and aggregated by the server. Comparing to traditional distributed learning solutions [18, 20], such mechanism could reduce communication overhead in large-scale IIoT network systems.

As summarized in Algorithm 2 with Option I (AutoGM_FL), let T be the number of global rounds, for each round $t \in [T]$, the server selects a subset S_t of devices randomly and broadcasts the global model $\mathbf{w}_t$ to the selected devices (Line 2). Then, every selected device performs E_g local updates using a batch of samples in each iteration, which is uniformly chosen from the local data (Line 6-10).

After E_g local iterations, the server aggregates the updated local model parameters $\{\mathbf{w}_k^t\}_{k \in S_t}$ from the devices S_t to

Algorithm 2 Optimization of FL and PFL with AutoGM

Input: $K, N, T, \eta, E_g, E_l, \mathbf{w}^0, \{\mathbf{v}_k^0\}_{k \in [N]}$

- 1: **for** each global round $t \in [T]$ **do**
- 2: Select a subset S_t from N devices at random, and broadcast $\mathbf{w}^t$ to them
- 3: **for** each client $k \in S_t$ **in parallel do**
- 4: $\mathbf{w}_k^t \leftarrow \mathbf{w}^t$
- 5: **Option I & II:**
- 6: **for** E_g local iterations, **do**
- 7: Sample mini-batch ξ from local dataset
- 8: $\mathbf{w}_k^t \leftarrow \mathbf{w}_k^t - \eta \nabla \tilde{F}_k(\mathbf{w}_k^t, \xi)$
- 9: **end for**
- 10: Sends $\mathbf{w}_k^t$ back to the server
- 11: **Option II (AutoGM_PFL):**
- 12: **for** E_l local iterations, **do**
- 13: Sample mini-batch ξ from local dataset
- 14: $\mathbf{v}_k \leftarrow \mathbf{v}_k - \eta \left(\nabla \tilde{F}_k(\mathbf{v}_k, \xi) + \theta (\mathbf{v}_k - \mathbf{w}^t) \right)$
- 15: **end for**
- 16: **end for**
- 17: **end for**
- 18: $\mathbf{w}^{t+1} \leftarrow \text{AutoGM}(\{\mathbf{w}_k^t\}_{k \in S_t})$ using Algorithm 1
- 19: **end for**
- 20: **Option I (AutoGM_FL):**
- 21: **return** $\mathbf{w}^T$
- 22: **Option II (AutoGM_PFL):**
- 23: **return** $\mathbf{w}^T, \{\mathbf{v}_k\}_{k \in [N]}$

produce a new global model. The local training and central aggregation steps are repeated until the global model converges.

Mean is widely used as an aggregation rule in FL approaches (e.g., FedAvg), which simply averages the parameters from the selected devices in each round, i.e.,

$$\mathbf{w}_{t+1} \leftarrow \frac{1}{K} \sum_{k \in S_t} \mathbf{w}_k^t. \quad (12)$$

However, Mean is vulnerable to malicious local parameters. As the breakdown point of Mean is $1/K$ [36], which means that even one of the selected devices is malicious, the resulting global model can significantly deviate from the original Mean.

Instead of Mean, we use our AutoGM to aggregate the local parameters (Line 18), where we optimize the AutoGM problem with Algorithm 1. According to the discussion of AutoGM in Section IV, $\mathbf{w}_{t+1}$ is a representative point of updated local model parameters $\{\mathbf{w}_k^t\}_{k \in S_t}$ that is resistant to malicious devices.

C. Extension to Personalized Federated Learning

We further explore the integration of AutoGM with Personalized Federated Learning (PFL) [23]. The goal of PFL is to jointly optimize the global objective in Equation (11) and a personalized local objective $\tilde{F}_k(\mathbf{v}_k)$ that aims to learn a model using local data of device k under the guidance of the global model. To this end, we incorporate a regularization term into the local objective to encourage the personalized models to be

close to the global model. Then the optimization problem for each device $k \in [N]$ is given by:

$$\begin{aligned} \min_{\mathbf{v}_k} \quad & h_k(\mathbf{v}_k) := \tilde{F}_k(\mathbf{v}_k) + \frac{\theta}{2} \|\mathbf{v}_k - \mathbf{w}^*\|^2 \\ \text{s.t.} \quad & \mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{N} \sum_{k \in [N]} \tilde{F}_k(\mathbf{w}), \end{aligned} \quad (13)$$

where vector $\mathbf{v}_k$ is the personal model parameter of node k , $\mathbf{w}^*$ is the vector that minimizes the global objective in Equation (11), and θ is a hyperparameter that controls the trade-off between local and global models. When $\theta = 0$, PFL is reduced to training individual local models; as θ grows large, it recovers global model optimization.

To solve the problem, we update the global model and local models in an alternating fashion as shown in Algorithm 2 with Option II (AutoGM_PFL). For each global round, each selected device first solves the local subproblem of Equation (11) approximately (Line 6-10), which follows the same update rule as AutoGM_FL. For the personalized local model, device k solves the global-regularized local objective $\min_{\mathbf{v}_k} h_k(\mathbf{v}_k)$ inexactly by running SGD for E_l iterations (Line 13-16). We also apply AutoGM to aggregate the parameters from the local devices to build a robust global model (Line 18).

We note that the local training of our AutoGM_PFL is the same as Ditto [23]. Although a more natural choice to solve Equation (13) is to first obtain $\mathbf{w}^*$ using AutoGM_FL, and then optimize each local objective under the guidance of $\mathbf{w}^*$. However, as noted by Li et al. [23], in non-convex settings, alternating schemes tend to achieve better solutions compared to fine-tuning starting from $\mathbf{w}^*$, especially in training-time attack scenarios.

D. Discussion

Overall, we note that AutoGM_FL and AutoGM_PFL have the following benefits for machine learning in IIoT systems:

- *Privacy*: As with typical FedAvg, both AutoGM_FL and AutoGM_PFL upload model parameters instead of local data over the network, which preserves privacy and avoids user data leakage.
- *Security*: The Byzantine-robust rule AutoGM is used as a secure aggregation protocol to alleviate the power of attackers by automatically excluding the extreme abnormal local parameters and re-weighting the rest according to a user-specified threshold of skewness. For AutoGM_FL, such a robust aggregation is critical to the security of the whole system, as all devices rely on a single global model. For AutoGM_PFL, AutoGM shapes the global model to better guild the training of personalized models.
- *Scalability*: During optimization, only a subset of devices is selected at each round and several steps of local training take place before global aggregation, which keeps the communication overhead low in large-scale IIoT networks and thus enhances the scalability of the systems.

A. Experimental Setup

We perform our experimental evaluation on two datasets. The FEMNIST dataset is an image classification dataset of handwriting from the LEAF framework [42], which has been commonly used to evaluate FL solutions [23]. The other dataset is provided by Bosch from an IIoT in a product assembly line of its manufacturing processes¹, which is a binary classification dataset.

Following our threat model described in Section III-B, we apply data poisoning attacks and model poisoning attacks to corrupt a random subset of devices. In the model poisoning scenario, we add Gaussian noise to the local parameters in the corrupted devices. The mean and variance of the noise is sampled from the uniform distribution $U_{[0,0.1]}$ for each corrupted device. For data poisoning attacks, we randomly shuffle the labels of all samples in each malicious device.

We implement the machine learning models in Pytorch, and simulate an IIoT system with one server and N edge devices². We compare our proposed AutoGM_FL and AutoGM_PFL with the following approaches:

- FedAvg [7]: The standard Federated Averaging aggregation approach that just calculates the weighted average of the parameters from local devices.
- FedProx [43]: An FL framework with a proximal term to help stabilize the method. It is a generalization and re-parametrization of FedAvg.
- RFA [21]: A robust aggregation approach that minimizes the weighted GM of the parameters from local devices.
- Ditto [23]: A PFL framework that provides personalization for local models.

The details of the datasets and the default hyperparameters of the corresponding models are listed in Table I. In consideration of training time, we only use subsets of the whole datasets. For FEMNIST, we treat each writer as a device and randomly sample subsets of all devices. As the data distributions vary among devices in the raw data, we regard this sampling process as non-i.i.d.. The raw data in FEMNIST is preprocessed using the popular benchmark LEAF [42], where the data on each device is partitioned into an 80% training set and a 20% test set. We train a CNN with two 5x5 convolution layers (the first with 32 channels, the second with 64, each followed by 2x2 max pooling), a fully connected layer with 126 units and ReLu activation, and a final softmax output layer.

For Bosch, we consider an i.i.d. partition where each local device has approximately the same amount of samples. We consider a Multi-Layer Perceptron (MLP) model with two fully connected layers with 512 units and ReLu activation, and a final softmax output layer.

For the FEMNIST dataset, we evaluate the performance of each approach with the accuracy. For the Bosch dataset, as the two classes are of very different sizes, we use the Matthews Correlation Coefficient (MCC) instead of accuracy. The MCC

¹<https://www.kaggle.com/c/bosch-production-line-performance/overview>

²Code is available at <https://github.com/lishenghui/AutoGM>

Dataset	# Classes	N	# Samples	i.i.d.	Model used	l_r	E_g	Batch size	K	T
Bosch	2	10	30,000	Yes	MLP	0.01	10	64	10	1000
FEMNIST	62	91	15,108	No	CNN	0.025	50	64	16	600

TABLE I: Experiment Summary

is given by:

$$MCC = \frac{(TP * TN - FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \quad (14)$$

where TP is the number of true positives, TN the number of true negatives, FP the number of false positives, and FN the number of false negatives. A coefficient of 1 represents a perfect prediction, and 0 indicates the performance is not better than random guessing.

B. Robustness Comparison

In Table II and III, we show that our proposed solutions, AutoGM_FL and AutoGM_PFL, are more robust by comparing the average test MCC score and accuracy over five random seeds.

First of all, when all datasets are clean, personalized approaches (i.e., Ditto, AutoGM_PFL) do not help to improve the performance on the Bosch dataset (Table II), but Ditto and AutoGM_PFL ($\lambda = 10^5 \times K$) both achieve higher accuracy than FedAvg and FedProx (Table III). This is because Bosch is an i.i.d dataset, then hence a single global model can utilize the information from all the data samples to a greater degree. FEMNIST, on the contrary, is a non-i.i.d dataset, which allows Ditto and AutoGM_PFL to build personalized models for local devices to better fit their own data distribution. It is also noticeable that when λ is small, i.e., $\lambda = K$, where K is the size of selected devices at each round, AutoGM_FL achieves slightly lower accuracy compared with other approaches on FEMNIST when all datasets are clean. An intuitive explanation is that down-weighting or excluding local parameters far from the AutoGM may result in losing correct information from devices, especially when the dataset is non-i.i.d.

On the contrary, a small λ is favorable when handling poisoning attacks. When $\lambda = K$, both AutoGM_FL and AutoGM_PFL successfully defend against up to 30% of nodes under model poisoning attacks and 50% malicious devices under data poisoning attacks while achieving better performance for both Bosch and FEMNIST. Table II and III also show that FedAvg, FedProx, and Ditto are all vulnerable to model poisoning attacks. Their MCC and accuracy significantly decreases as the presence of model poisoning attacks even when only 10% of the devices are malicious. The reason behind this is that they all aggregate the parameters using Mean, which could be easily driven away by the malicious updates as we discussed in Section IV-B.

Fig. 3a and 3b show the performance of the robust approaches versus training rounds under model poisoning attacks and data poisoning attacks, respectively. When $\lambda = K$, both AutoGM_FL and AutoGM_PFL have steadily increasing performances in terms of MCC and accuracy during the training process, which implies that they successfully overcome the

impact of malicious devices. When $\lambda = 10^5 \times K$, the curves of AutoGM_FL on Bosch almost coincide with RFA, as their aggregation rules are both reduced to the naive GM when all the local datasets have the same size. We also note that the MCC score fluctuates over training rounds when λ is large in AutoGM_FL and AutoGM_PFL (Fig. 3a left), one explanation for this is that the devices are selected randomly for local training at each round, making it possible that more malicious devices are selected than benign ones at some point. In this situation, AutoGM cannot down-weight all the malicious updates if λ is large. On the contrary, when λ is small, AutoGM handles it better, as the curves are more stable, leading to relatively higher scores when $\lambda = K$. For model poisoning attack on FEMNIST, AutoGM_FL is still robust, while RFA fails after 500 rounds of training (Fig. 3a right). In this case, AutoGM_FL benefits from our scaled formulation in Section V-A, which aligns the aggregation rule with GM even the devices have different dataset sizes, and preserves the robustness property.

C. Impact of λ

Next we study the impact of the hyperparameter λ , which controls the L_1 -skewness between local parameters with their Euclidean distance to AutoGM. We change the value of λ in the range of $[10^{-2} \times K, 10^5 \times K]$, where K is the number of devices selected in each round. We compare the accuracy of AutoGM_FL and AutoGM_PFL on FEMNIST under data poisoning attacks with 50% of malicious devices. Fig. 4-(a) shows the average accuracy and the gap between 25-th and 75-th percentile over all benign devices. The figure shows that λ has a significant impact on the performance of both approaches. Specifically, both AutoGM_FL and AutoGM_PFL reach their peak performances when $\lambda = K$. Overall, AutoGM_PFL performs better than AutoGM_FL in terms of average accuracy with all values of λ .

D. Impact of θ

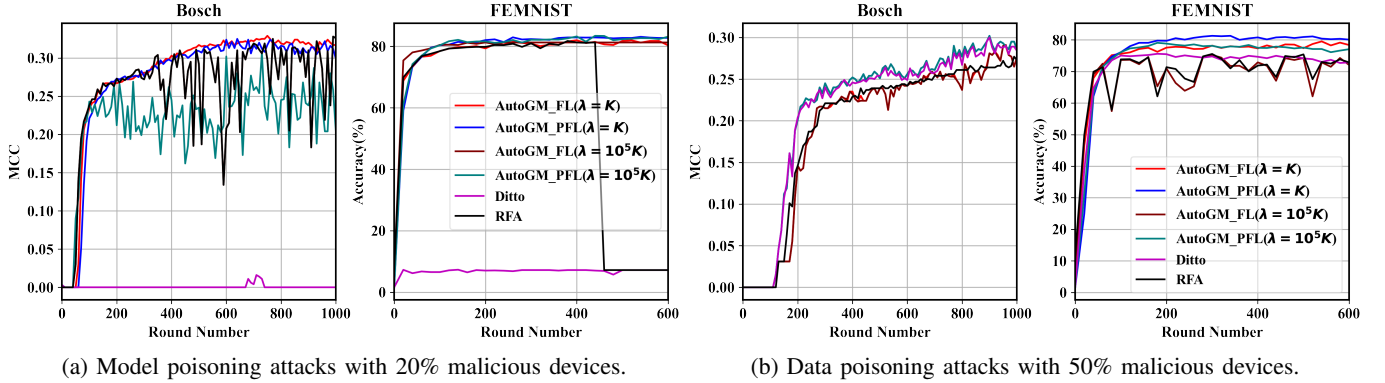
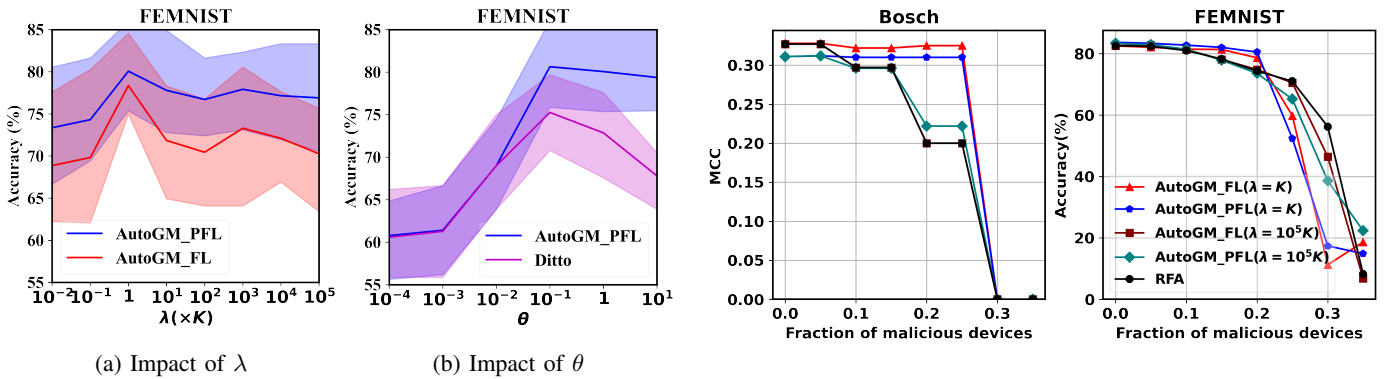
For PFL-based approaches (AutoGM_PFL and Ditto), we further study the impact of hyperparameter θ , which controls the similarity between the personalized model and the global model. We conduct experiments on FEMNIST with 50% of devices under data poisoning attacks and set the value of θ in the range of $[10^{-4}, 10]$. We show the average accuracy and the gap between 25-th and 75-th percentile of AutoGM_PFL and Ditto in Fig. 4-(b). The performance of both approaches increases as the value of θ increases and reaches their peaks when $\theta = 10^{-1}$, which coincides with the results in [23]. However, when θ further increases, the accuracy slightly decreases as all the local models are forced to be more and more similar to the global model and gradually lose the advantage of

Bosch	Clean	Model Poisoning			Data Poisoning		
Fraction of malicious	-	10%	20%	30%	10%	30%	50%
FedAvg	0.324 (0.001)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.324 (0.007)	0.31 (0.007)	0.258 (0.017)
FedProx	0.324 (0.002)	0.003 (0.006)	0.0 (0.0)	0.003 (0.006)	0.324 (0.009)	0.314 (0.004)	0.259 (0.025)
RFA	0.324 (0.003)	0.318 (0.003)	0.309 (0.017)	0.121 (0.149)	<i>0.326 (0.005)</i>	0.315 (0.007)	0.265 (0.017)
Ditto	0.31 (0.01)	0.01 (0.02)	0.0 (0.0)	0.0 (0.0)	0.32 (0.01)	0.31 (0.01)	0.29 (0.01)
AutoGM_FL ($\lambda = K$)	0.324 (0.003)	0.32 (0.003)	0.318 (0.005)	0.283 (0.019)	<i>0.326 (0.005)</i>	0.316 (0.007)	0.265 (0.017)
AutoGM_FL ($\lambda = 10^5 K$)	0.324 (0.003)	0.318 (0.003)	0.309 (0.017)	0.121 (0.149)	<i>0.326 (0.005)</i>	0.315 (0.007)	0.265 (0.017)
AutoGM_PFL ($\lambda = K$)	0.31 (0.0)	0.31 (0.01)	0.3 (0.01)	0.28 (0.02)	0.32 (0.01)	0.31 (0.01)	0.29 (0.01)
AutoGM_PFL ($\lambda = 10^5 K$)	0.31 (0.0)	0.3 (0.01)	0.25 (0.03)	0.07 (0.11)	0.32 (0.01)	0.31 (0.01)	0.29 (0.01)

TABLE II: Averaged (standard deviation) test MCC on Bosch in model poisoning and data poisoning attacks.

FEMNIST	Clean	Model Poisoning			Data Poisoning		
Fraction of malicious	-	10%	20%	30%	10%	30%	50%
FedAvg	83.47 (0.22)	7.43 (0.15)	7.23 (0.53)	7.32 (0.33)	78.95 (1.19)	67.05 (3.93)	55.51 (4.13)
FedProx	83.62 (0.36)	7.18 (0.25)	7.17 (0.36)	7.3 (0.39)	79.7 (0.77)	62.82 (6.07)	49.53 (3.68)
RFA	82.81 (0.46)	81.79 (0.58)	50.98 (35.68)	7.43 (0.24)	82.56 (0.38)	79.25 (1.36)	74.82 (1.36)
Ditto	84.5 (0.34)	7.28 (0.09)	7.14 (0.31)	6.92 (0.61)	81.45 (0.51)	77.11 (0.62)	73.44 (0.45)
AutoGM_FL ($\lambda = K$)	82.2 (0.41)	81.51 (0.83)	81.55 (0.7)	81.53 (0.8)	81.85 (0.51)	81.44 (0.89)	79.61 (0.7)
AutoGM_FL ($\lambda = 10^5 K$)	82.98 (0.56)	81.99 (0.46)	81.3 (1.18)	21.36 (27.76)	82.11 (0.63)	78.72 (1.71)	70.57 (2.28)
AutoGM_PFL ($\lambda = K$)	84.02 (0.76)	83.55 (0.91)	83.66 (0.6)	68.16 (30.84)	83.73 (0.78)	82.8 (0.65)	81.38 (1.04)
AutoGM_PFL ($\lambda = 10^5 K$)	84.53 (0.62)	83.82 (0.44)	67.57 (30.18)	5.41 (1.16)	83.82 (0.46)	81.4 (0.72)	77.42 (0.96)

TABLE III: Averaged (standard deviation) test accuracy (%) on FEMNIST in model poisoning and data poisoning attacks.

Fig. 3: Impact of λ and θ on test accuracy.Fig. 4: Impact of λ and θ on test accuracy.

personalization. Our AutoGM_PFL results in higher accuracy than Ditto when $\theta \geq 10^{-2}$, showing the superiority of our robust aggregation rule.

Fig. 5: Model poisoning attack with collusion.

E. Collusion Attacks

We further conduct the experiments on model poisoning attack with collusion. In this experiments, malicious devices collude by sending the same corrupted model parameters to the server. Sending the same parameters has a higher chance to negatively impact the global model compared to, for example,

randomly selecting corrupted parameters. Recall that when $\lambda = K$, both `AutoGM_FL` and `AutoGM_PFL` can tolerate up to 30% of devices under model poisoning attacks without collusion. Fig. 5 shows that the tolerance decreases in the presence of collusion. Specifically, all the MCC scores drop to zero when there are 30% malicious devices on Bosch, and the accuracy on FEMNIST is dramatically lower compared to the non-colluding model poisoning attacks in Table II and III. As GM is determined by the majority, colluding devices could easily dominate the aggregation with the same parameters, as long as most of the selected devices are malicious. Nevertheless, with $\lambda = K$, our methods still outperform RFA with up to 20% malicious and colluding devices.

VII. CONCLUSIONS

In this paper, we propose `AutoGM`, a flexible Byzantine-robust aggregation rule to handle model poisoning attacks and data poisoning attacks in FL-IIoT systems. `AutoGM` seamlessly integrates with both standard FL and personalized FL. Our experimental evaluation shows that our solutions achieve better robustness than other state-of-the-art methods. In particular, they sustains high performance even when 30% of the nodes perform model or 50% of the nodes data poisoning attacks. In future work, we will explore the possibility of using `AutoGM` as an aggregation rule in federated generative adversarial training to further enhance the robustness of the global and local models. Also, we will consider more types of attacks, such as false data injection attacks and backdoor attacks in IIoT applications.

APPENDIX A

PROOF OF THEOREM 1

Proof. Note that $\{\|z - z_k\|\}_{k \in S}$ is given, we sort the elements in increasing order and denote them by a vector $\mathbf{d}$ where $d_k = \|z - z_k\|$. Then the Lagrangian function of Equation (2) is

$$\mathbb{L} = \alpha^\top \mathbf{d} + \frac{\lambda}{2} \|\alpha^\top\|_2^2 - \alpha^\top \beta - \eta(\alpha^\top \mathbf{1} - 1), \quad (15)$$

where β and η are the Lagrangian multipliers. Then the following Karush–Kuhn–Tucker (KKT) conditions hold:

$$\partial_\alpha \mathbb{L}(\alpha, \beta, \eta) = 0, \quad (16)$$

$$\alpha^\top \mathbf{1} - 1 = 0, \quad (17)$$

$$\alpha \geq 0, \quad (18)$$

$$\beta \geq 0, \quad (19)$$

$$\alpha_k \beta_k = 0, \forall k = 1, 2, \dots, K. \quad (20)$$

According to Equation (16), we have:

$$\alpha_k = \frac{\beta_k + \eta - d_k}{\lambda}. \quad (21)$$

Since $\beta_i \geq 0$, we discuss the following :

- 1) When $\beta_k = 0$, we have $\alpha_k = \frac{\eta - d_k}{\lambda} \geq 0$, which further indicates $\eta - d_k \geq 0$.
- 2) When $\beta_k > 0$, from the condition $\alpha_k \beta_k = 0$, we have $\alpha_k = 0$.

Therefore, the optimal solution to Equation (1) is given by:

$$\alpha_k = \frac{1}{\lambda} [\eta - d_k]_+, \quad (22)$$

where $[\cdot]_+ = \max(0, \cdot)$.

Since $\{d_k\}_{k \in [K]}$ appears in the increasing order, thus we have $\alpha_1 \geq \alpha_2, \dots, \geq \alpha_K$. Suppose $\alpha_p > 0$ and $\alpha_{p+1} \leq 0$, then the support set is $S_p = \{k\}_{k \in [p]}$. Further more, given that $\sum_{k=1}^p \alpha_k = 1$, we can get:

$$\eta = \frac{\sum_{k=1}^p d_k + \lambda}{p}. \quad (23)$$

Plugging Equation (23) into Equation (22) yields Equation (6).

Next, since if $\alpha_k \geq 0$, we have $\eta \geq d_k$, which means

$$\frac{\sum_{k=1}^p d_k + \lambda}{p} \geq \max(\{d_k\}_{k \in [p]}). \quad (24)$$

The equation above can be readily converted to Equation (5). This completes the proof. $\square$

REFERENCES

- [1] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, 2013.
- [2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011.
- [3] M. Pech, J. Vrchota, and J. Bednár, "Predictive maintenance and intelligent sensors in smart factory," *Sensors*, 2021.
- [4] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: Challenges, solutions and opportunities," *IEEE Network*, 2018.
- [5] M. S. Mahdavejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: A survey," *Digital Communications and Networks*, 2018.
- [6] Y. Wu, Z. Wang, Y. Shi, and J. Hu, "Enabling on-device cnn training by self-supervised instance filtering and error map pruning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.
- [7] 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*, 2017.
- [8] M. Hao, H. Li, X. Luo, G. Xu, H. Yang, and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence," *IEEE Transactions on Industrial Informatics*, 2019.
- [9] Y. Yu, S. Liu, P. Yeoh, B. Vucetic, and Y. Li, "Layerchain: A hierarchical edge-cloud blockchain for large-scale low-delay iiot applications," *IEEE Transactions on Industrial Informatics*, 2020.
- [10] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-iiot: A robust federated malware detection architecture in industrial iiot," *IEEE Transactions on Industrial Informatics*, 2020.
- [11] S. Messaoud, A. Bradai, O. B. Ahmed, P. Quang, M. Atri, and M. S. Hossain, "Deep federated q-learning-based network slicing for industrial iiot," *IEEE Transactions on Industrial Informatics*, 2020.
- [12] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th {USENIX} Security Symposium*, 2020.
- [13] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun, "Understanding distributed poisoning attack in federated learning," in *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2019, pp. 233–239.
- [14] Z. Xiao, H. Fang, and X. Wang, "Anomalous iiot sensor data detection: An efficient approach enabled by nonlinear frequency-domain graph analysis," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3812–3821, 2020.
- [15] F. Ang, L. Chen, N. Zhao, Y. Chen, W. Wang, and F. R. Yu, "Robust federated learning with noisy communication," *IEEE Transactions on Communications*, 2020.
- [16] 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*, 2017.

- [17] H. Zhou, C. She, Y. Deng, M. Dohler, and A. Nallanathan, "Machine learning for massive industrial internet of things," *arXiv preprint arXiv:2103.08308*, 2021.
- [18] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 2017.
- [19] C. Xie, O. Koyejo, and I. Gupta, "Generalized byzantine-tolerant sgd," *arXiv preprint arXiv:1802.10116*, 2018.
- [20] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018.
- [21] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.
- [22] Q. Wu, K. He, and X. Chen, "Personalized federated learning for intelligent iot applications: A cloud-edge based framework," *IEEE Open Journal of the Computer Society*, 2020.
- [23] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International Conference on Machine Learning*, 2021.
- [24] P. Yu, A. Kundu, L. Wynter, and S. H. Lim, "Fed+: A unified approach to robust personalized federated learning," *arXiv preprint arXiv:2009.06303*, 2020.
- [25] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "Iotdefender: A federated transfer learning intrusion detection framework for 5g iot," in *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*. IEEE, 2020.
- [26] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, 2020.
- [27] V. Rey, P. M. S. Sánchez, A. H. Celdrán, G. Bovet, and M. Jaggi, "Federated learning for malware detection in iot devices," *arXiv preprint arXiv:2104.09994*, 2021.
- [28] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of iiot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8475–8484, 2021.
- [29] W. Zhang, D. Yang, W. Wu, H. Peng, H. Zhang, and X. S. Shen, "Spectrum and computing resource management for federated learning in distributed industrial iot," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.
- [30] Y. Lu, X. Huang, and K. Z. et al., "Communication-efficient federated learning for digital twin edge networks in industrial iot," *IEEE Trans. Ind. Informatics*, 2021.
- [31] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial iot and smart city services," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.
- [32] S. Otoum, N. Guizani, and H. Mouftah, "Federated reinforcement learning-supported ids for iot-steered healthcare systems," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.
- [33] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International Conference on Artificial Intelligence and Statistics*, 2020.
- [34] M. Song, Z. Wang, Z. Zhang, Y. Song, Q. Wang, J. Ren, and H. Qi, "Analyzing user-level privacy attack against federated learning," *IEEE Journal on Selected Areas in Communications*, 2020.
- [35] N. Konstantinov and C. Lampert, "Robust learning from untrusted sources," in *International conference on machine learning*, 2019.
- [36] H. P. Lopuhaa, P. J. Rousseeuw et al., "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *The Annals of Statistics*, 1991.
- [37] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," *arXiv preprint arXiv:2003.02133*, 2020.
- [38] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation," in *Uncertainty in Artificial Intelligence*. PMLR, 2020, pp. 261–270.
- [39] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*. PMLR, 2019.
- [40] M. B. Cohen, Y. T. Lee, G. Miller, J. Pachocki, and A. Sidford, "Geometric median in nearly linear time," in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 2016.
- [41] 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*, 2019.
- [42] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: A benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, 2018.
- [43] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, 2020.



Shenghui Li received the B.S. degree from Xidian University, Xi'an, China, in 2017, and the M.S. degree from Sun Yat-sen University, Guangzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Information Technology, Uppsala University, Uppsala, Sweden. His current research interests include federated learning, distributed optimization, and statistical machine learning.



Edith Ngai is currently an Associate Professor in Department of Electrical and Electronic Engineering, The University of Hong Kong. Before joining HKU in 2020, she was an Associate Professor in the Department of Information Technology, Uppsala University, Sweden. Dr. Ngai was a guest researcher at Ericsson Research Sweden in 2015-2017. Previously, she has conducted research in Imperial College London, Simon Fraser University, Tsinghua University, and UCLA. Her research interests include Internet-of-Things, machine learning, data analytics, and smart cities. Dr. Ngai is a VINNMER Fellow (2009) awarded by Swedish Governmental Research Funding Agency VINNOVA. She led the "Green IoT" project in Sweden, which was named on the IVA's 100-list from the Royal Swedish Academy of Engineering Sciences in 2020. She is currently an Area Editor of IEEE Internet of Things Journal and an Associate Editor of IEEE Access and IEEE Transactions on Industrial Informatics.



(TOSN).

Thiemo Voigt received the Ph.D. degree from Uppsala University, Sweden, in 2002. He is currently a Professor of computer science with the Department of Information Technology, Uppsala University. He also leads the Networked Embedded Systems Group, RISE Computer Science. His current research interests include system software for embedded networked devices and the Internet of Things. His work has been cited more than 14 500 times. He is a member of the editorial board for the IEEE INTERNET OF THINGS and ACM Transactions on Sensor Networks