



Article

Enhancing Robustness within the Collaborative Federated Learning Framework: A Novel Grouping Algorithm for Edge Clients

Zhi-Yuan Su ¹, I-Hsien Liu ² , Chu-Fen Li ³, Chuan-Kang Liu ^{4,*} and Chi-Hui Chiang ⁵ 

¹ Department of Multimedia and Game Development, Chia-Nan University of Pharmacy and Science, No. 60, Sec. 1, Erren Rd., Rende Dist., Tainan City 717301, Taiwan; zysu@mail.cnu.edu.tw

² Department of Electrical Engineering, Institute of Computer and Communication Engineering, National Cheng Kung University, No. 1, University Rd., East Dist., Tainan City 701401, Taiwan; ihliu@cans.ee.ncku.edu.tw

³ Department of Finance, National Formosa University, No. 64, Wunhua Rd., Huwei Township, Yunlin County 632301, Taiwan; chufenli@gmail.com

⁴ Department of Artificial Intelligence and Computer Engineering, National Chin-Yi University of Technology, No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan

⁵ Department of Information Management, Chia-Nan University of Pharmacy and Science, No. 60, Sec. 1, Erren Rd., Rende Dist., Tainan City 717301, Taiwan; cscott@mail.cnu.edu.tw

* Correspondence: chgliu090210@gmail.com

Abstract: In this study, we introduce a novel collaborative federated learning (FL) framework, aiming at enhancing robustness in distributed learning environments, particularly pertinent to IoT and industrial automation scenarios. At the core of our contribution is the development of an innovative grouping algorithm for edge clients. This algorithm employs a distinctive ID distribution function, enabling efficient and secure grouping of both normal and potentially malicious clients. Our proposed grouping scheme accurately determines the numerical difference between normal and malicious groups under various network scenarios. Our method addresses the challenge of model poisoning attacks, ensuring the accuracy of outcomes in a collaborative federated learning framework. Our numerical experiments demonstrate that our grouping scheme effectively limits the number of malicious groups. Additionally, our collaborative FL framework has shown resilience against various levels of poisoning attack abilities and maintained high prediction accuracy across a range of scenarios, showcasing its robustness against poisoning attacks.

Keywords: federated learning; poisoning attacks; grouping scheme



Citation: Su, Z.-Y.; Liu, I.-H.; Li, C.-F.; Liu, C.-K.; Chiang, C.-H. Enhancing Robustness within the Collaborative Federated Learning Framework: A Novel Grouping Algorithm for Edge Clients. *Appl. Sci.* **2024**, *14*, 3255. <https://doi.org/10.3390/app14083255>

Academic Editor: Mirosław Klinkowski

Received: 27 February 2024

Revised: 10 April 2024

Accepted: 10 April 2024

Published: 12 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, Machine Learning has provided intelligent online automation control in various fields, especially in IoT and industrial automation. In such environments, there are many sensors deployed for collecting sensed data. However, the sensed data of these sensors is often private and need to be protected. Yet, in online collecting of these data sets, their large volume in transit make them easily vulnerable to hijacking by malicious users, leading to data leakage, which is unacceptable to both users and the industry managers. Hence, a new paradigm, the federated learning (FL) framework, is proposed, which is a secure machine learning framework ensuring privacy protection and providing a flexible architecture [1]. In such an emerging learning paradigm, the edge client executes a specified function that is different from the previous one, just transferring its local dataset to the central server. In the FL framework, the edge client trains its learning model locally using its own dataset. Then the server collects all the learning model weights from the edge clients and generates a global learning model based on an FL algorithm. Upon receiving this global model, the edge client replaces its local learning model with this newly received one. Then it trains this model again and repeats the aforementioned processes. This workflow is also called the single-global-model paradigm. This paradigm has been

suggested to be deployed in several fields, including IoT applications [2–4], industry applications [5], network applications [6] and so on. While FL offers significant advantages in distributed environments, it concurrently faces substantial security challenges [7,8], particularly from malicious clients. Among these, the poisoning attack is a critical threat. In a poisoning attack, malicious clients intentionally skew the training data or manipulate the learning process by injecting false information or models, which results in a significantly compromised global model. Such compromised models can lead to erroneous decisions or actions, undermining the integrity and reliability of the entire FL system. Recently, numerous studies have concentrated on developing robust defense mechanisms to detect and mitigate the effects of these attacks. These include techniques for identifying anomalous data contributions and enhancing the aggregation algorithms to resist the influence of malicious updates.

Recently, the authors of [9] proposed an idea for breaking the single-global-model framework, called a collaborative federated learning (FL) framework. The key idea in this learning framework is to divide the edge clients into several groups. There are two types of groups in this framework, the normal group and the malicious group. The latter one involves the malicious clients. Each group performs the fundamental FL mechanism and generates its corresponding global model. Each edge client receives all global models from all the groups and employs them to predict the output prediction via a voting strategy. Thus, the edge client generates the final output based on the majority consensus of the global models. In this collaborative FL framework, grouping enables the segregation of edge clients into manageable subsets. This becomes crucial in large-scale deployments. By categorizing nodes into groups, isolating potentially malicious clients and minimizing their impact on the prediction output can be achieved in a collaborative FL framework. This approach is particularly effective in mitigating the risk of a malicious node influencing the global learning outcome. It is observed that the accuracy of this voting scheme varies significantly according to the ratio of the number of the normal groups to that of malicious ones.

Based on the above observation, we here develop our proposed collaborative FL framework with a robust grouping scheme based on our early work [10]. In our proposed framework, each edge client initially obtains an ID assigned by the administrator. These IDs form a specified distribution. In essence, the malicious user fakes an edge client and sends its local model with a randomly chosen ID. With the feature of this ID distribution, the grouping administrator can easily distinguish whether an edge client is legitimate or not. Hence, we can create several specified groups for those potentially malicious clients. Through this manipulating ID distribution feature design, even if the total number of groups in this collaborative federated learning is low, the proposed grouping scheme can continue to ensure a high accuracy even as the number of malicious edge clients increases.

2. Related Works

Nowadays, privacy protection provided by FL is more robust and efficient than was the case with the traditional ML model, while FL still ensures a similar prediction accuracy. As stated in a survey study [11], recent security defense instances [12–18] are also being developed based on the FL framework, including federated learning-based intrusion detection systems and anomaly detection. In the Internet of Things (IoT) Industry 4.0, the authors in [14] proposed a collaborative intrusion detection system (IDS) in which there are filters performing a deep neural network (DNN) and a central server collecting the filters' DNN parameters to generate a global model. Recently, researchers have started to employ FL (federated learning) technology to achieve privacy protection and high accuracy in Android malware detection and classification. In [19], the authors use federated learning, combining data from multiple users, to improve malware detection and ensure privacy preservation. The authors in [20] propose another framework based on a combination of semi-supervised machine learning and federated learning. Its focus is on maintaining user privacy and it employs a semi-supervised machine learning technique that reportedly

improves classification accuracy. The authors in [21] propose LiM, a malware classification framework that leverages the power of FL to detect and classify malicious apps in a privacy-respecting manner. The methodology in [22] is distinctive because it integrates federated learning (FL) with a novel classification model to protect user privacy while effectively identifying malware. These works employ an FL framework to enable distributed android clients to collaboratively train a comprehensive Android malware detection or classification in a privacy-preserving manner. However, if FL faces cybersecurity attacks, all applications using this framework might also be exposed to several security risks, such as private user data leakage and reduced accuracy. This study aims to propose a new robust FL framework resistant to cybersecurity attacks. Our proposed FL framework can ensure that FL can resist multiple attacks and maintain high accuracy under various network scenarios.

The aim of cybersecurity attacks in the FL framework is to make the global model unstable and incapable of making accurate predictions. There are typically three main targets that the attacker might focus on. The first target is the local data set. The second is the local model maintained by the local edge client. The final one is the central server. The biased global model, once distributed to all edge clients, leads to incorrect label predictions for the inputs locally. Specifically, federated learning with an easy FL algorithm, e.g., FedAvg, is highly vulnerable to poisoning attacks, even if only one fake client is involved in the FL framework.

The famous case of the Byzantine problem [23] can cause fatal damage to the quality of the global model in an FL framework applying a FedAvg scheme. In order to confront such a poisoning attack, some new FL algorithms are proposed, like Krum [24] and Trimmed-mean [25]. Filtering possibly malicious model updates is an efficient method to prevent the global model from being influenced by poisoning attacks. Recently, a novel cybersecurity defense algorithm [9] for FL frameworks, called FLCert, was proposed. The proposed key idea divided the edge clients participating in the FL framework into several groups. According to their grouping methodology, there are two variants of grouping schemes, FLCert-P and FL Cert-D. In each group, each edge client trains its local model as a usual FL framework. Instead of sending local model parameters to the central server, the edge clients send local model parameters to their corresponding group leaders. These group leaders execute a similar job to that of the central server in a traditional FL framework, generating the group's global model with an FL algorithm. As a result, several global models are generated in such a learning paradigm. Then a voting scheme among all global models is applied to make the final prediction result for inputs. We call this framework a collaborative FL framework, which actually provides a new and efficient FL cybersecurity defense framework against poisoning attacks. However, though experimental result indicate more groups can ensure higher accuracy for the input, for a small-scale learning architecture on the other hand, the group number cannot increase indefinitely. Furthermore, there is a risk that the malicious users might manipulate the voting results because they can create many malicious edge clients. This significantly increases the likelihood that many groups will contain at least one malicious client, consequently leading to an increase in the number of the malicious groups. In this paper, we try to design a new grouping method that can mitigate the risk associated with this issue. Figure 1 shows the workflow of the basic collaborative FL framework.

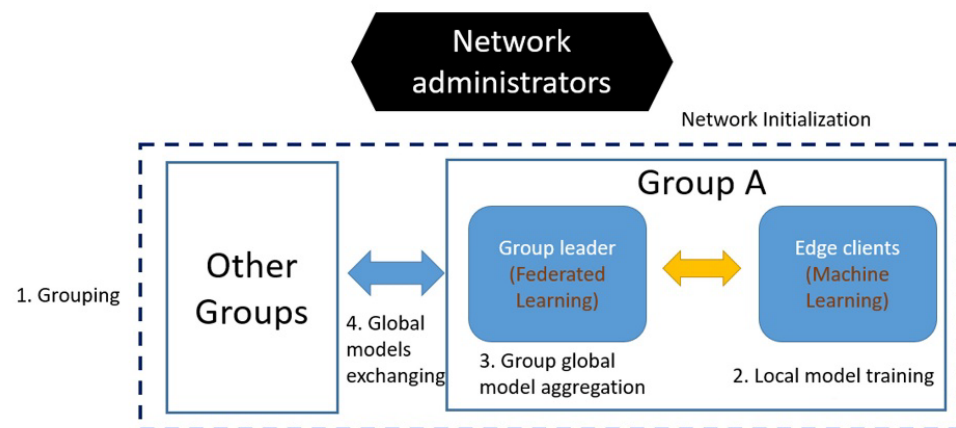


Figure 1. The workflow in a collaborative FL framework.

3. Proposed Collaborative FL Framework

In this work, a new grouping method in a collaborative FL framework is proposed based on an ID-distribution feature. We firstly introduce our study architecture and our adversary model. Next, we explain our proposed grouping scheme in our collaborative FL framework.

3.1. Network Architecture

In our network architecture, there are many sensors and the edge client collects the sensed data within the sensing range. Edge clients include both normal and malicious ones. n_n and n_m denote their respective numbers. All clients are divided into groups according to the grouping scheme. Total groups, N , contain N_n and N_m subgroups. N_n represents normal groups while N_m represents malicious ones. In each group, all local clients conform to the operations of FL. In order to execute the federated learning mechanism within each group, a group leader should be selected to be in charge of executing the FL algorithm. In our work, each group leader adopts the base FL algorithm, FedAvg, to generate its global model. A central server executes ID assignment and the grouping scheme.

3.2. Adversary Model

In our study, the capacity of one adversary is defined as follows.

1. The adversary is not allowed to join the initial registration process but is able to forge a malicious model with a randomly selected ID. Hence, the adversary can deviate the global model of a specified group in an unexpected direction. Once this event occurs, we also can say this group is compromised by the adversary. We call this compromised group the malicious one.
2. Inside each group, the adversary can overhear the messages in transit between the clients and local group leader and remove them.
3. The adversary cannot decrypt the encrypted messages in transit between the clients and the central server in time. Hence, the privacy of clients' IDs will not be jeopardized.

3.3. Basic Workflow

Figure 2 illustrates our basic workflow in a collaborative FL framework. Initially, the central server accepts the registration of all edge clients through the secure channel. All legitimate edge clients receive their ID assignment from the server. No ID information will be stored anywhere except for each edge client storing their own ID. Then, the server awaits the edge clients' grouping requests. Upon receiving the grouping requests, the server collects the participating clients' ID and divides them into corresponding groups through a grouping scheme. Although malicious users do not join the initial registration, they can forge a client's profile with a randomly chosen ID. In order to avoid ID leakage, each edge client has to encrypt a message with the public key sent by the server before sending out ID

information. Hence, the server can decrypt ID messages and execute the grouping scheme. Based on our adversary model, the adversary can overhear this encrypted message but cannot decrypt it in time. According to the grouping scheme, the groups are created and the edge client with the largest ID number is selected as the group leader in a given group. Then, all groups execute the FL mechanism. In each group, each edge client receives the initial model from the group leader and trains this local learning model with the local dataset. According to the principle of FL, all edge clients send their trained model to the corresponding group leader which aggregates all local models to generate a global model using a base FL algorithm. By now, all group leaders have their corresponding global models. Next, all group models are sent out to all participating clients. All participating clients can make output predictions through the received global models, and the output predicted by the majority of global models is considered the final output. This is the essence of the voting mechanism. Of course, these global models include normal and malicious ones. Through our proposed collaborative FL framework, the edge client is capable of making accurate output predictions even in the presence of numerous malicious clients.

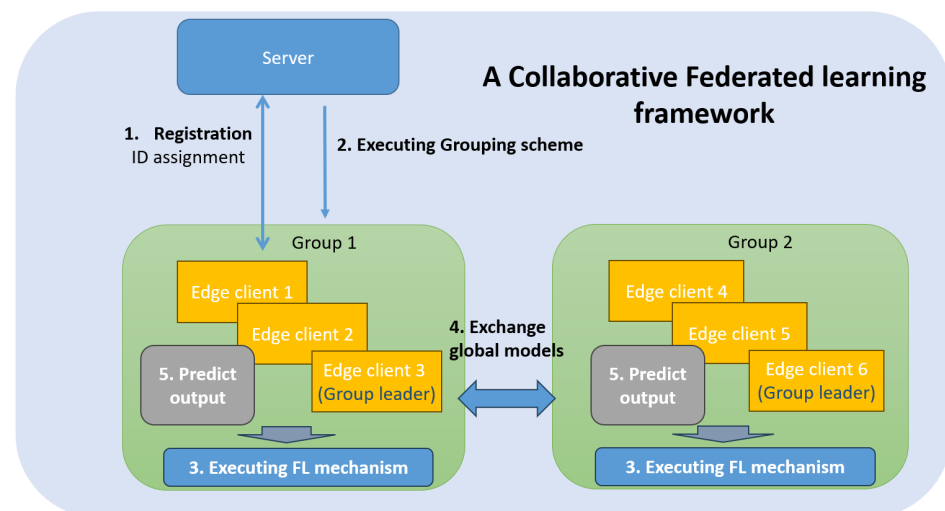


Figure 2. The workflow in our proposed collaborative FL framework.

The following subsections describe the working steps in the whole workflow.

3.4. ID Assignment

Initially, all legitimate edge clients should register with the server and receive IDs from the server. The malicious user can assign a randomly chosen ID to the malicious edge clients. Then, it also can join the subsequent grouping procedure. Therefore, in order to prevent the malicious users from invading our collaborative FL framework, the ID assignment rule is key. Because the central server does not store ID information, it cannot tell the difference between normal and malicious ones. A specified ID-distribution feature can resolve this dilemma. In our ID assignment, ID is an integer which is selected from the number of a specified distribution. Here, a normal distribution is utilized. Hence, we generate ID from the normal distribution. Basically, the server stores the ID distribution features instead of the IDs themselves. Hence, the IDs of legitimate users and malicious users are readily differentiated from the attributes of the ID distribution. The server only retains information about this ID distribution feature, which helps prevent the leakage of clients' ID information. The range of the ID number depends on the bit size storing this ID number. A longer bit length is generally preferred for security reasons. To the best of our knowledge, this ID assignment with the specified ID distribution feature in a collaborative FL framework is proposed here for the first time. Here, we denote the normal distribution, as $N(\lambda, \sigma)$, where λ is the mean and σ is the standard deviation. To avoid the selection of non-positive ID numbers, we set λ to be at least 4σ , typically preferring it to be more than

6σ . To easily distinguish if an ID is malicious or not, we also favor a narrow range of ID distribution, meaning σ should be as small as possible. Therefore, with a long bit length and a narrow range for legitimate ID distribution, the likelihood that one ID that the malicious user selects randomly collides with a legitimate ID is expected to be very low. Later, we will discuss this further when computing the number of normal and malicious groups.

3.5. Proposed Grouping Scheme

3.5.1. Distinguishing between Edge Clients

After ID assignment, the server performs the grouping scheme. First, the server collects the ID information of all participating clients and divides them into groups based on the grouping scheme. In the end of this grouping process, there are N groups in total, which are composed of normal and malicious ones, respectively N_n and N_m . A malicious group indicates that there is at least one malicious client in it while a normal one contains no malicious clients. There are k_n and k_m edge clients in normal and malicious groups, respectively. the aim of our grouping scheme is to classify all edge clients into normal groups or malicious ones. The number of normal groups is higher than that of malicious ones. In order to execute the grouping scheme, the first step is to distinguish malicious clients from all participating clients. Then the server executes the grouping scheme for both types of clients separately. The next paragraph explains our proposed distinguishing principle.

The distinguishing principle: If client's ID falls in the range of $\lambda \pm 4\sigma$, this client is regarded as a normal client.

According to the definition of normal distribution, 99.99% of samples fall in this range, $\lambda \pm 4\sigma$. Hence, the grouping administrator can easily tell one client's type on the basis of two parameters, λ , 4σ . Later, we discuss the impact of a false positive rate on our grouping scheme.

3.5.2. The Number of Normal and Malicious Groups

After classifying all participating clients into two categories, normal and malicious ones, the server begins to execute the grouping scheme for both client categories. However, according to the rules of the voting game, the player with the most votes wins the game. Hence, the grouping scheme should consider some constraints. First, we should ensure that the number of normal groups is larger than the number of malicious ones. Hence, the following equations should hold while grouping clients. Equation (1) represents the sum of clients in both categories being equal to the total number of clients. Equation (2) expresses the constraints we stated above.

$$k_m N_m + k_n N_n = n \quad (1)$$

$$N_m < N/2, N_m + N_n = N \quad (2)$$

Although the above constraints mainly ensure $N_n > N_m$, misclassification between two categories of edge clients may occur. This could potentially result in some normal groups being misclassified as malicious ones. Unless the above issues is resolved, the result of the grouping scheme may not be reliable. So, before grouping participating edge clients, we should pre-compute an adequate number of normal and malicious groups. In order to pre-compute the number of normal and malicious groups, we consider the impact of a false positive rate of classifying edge clients and the impact of prediction accuracy of global models on our grouping scheme, respectively. We can then finally obtain the minimum number of normal groups required, and the maximum number of malicious groups that can be tolerated.

The Impact of a False Positive Rate of Classifying Edge Clients

According to our design, a valid ID is selected from a normal distribution with mean λ and standard deviation σ . We also assume that a malicious user may forge many

malicious clients with randomly selected IDs. The grouping administrator distinguishes all participating clients based on the distinguishing principle. Hence, a malicious client with an ID falling in this range, $\lambda \pm 4\sigma$, is misjudged as a normal one, which is called a false positive case. Now, we try to compute the probability of this event. First, we assume that D denotes the total number of all IDs. According to our ID assignment, D is significantly larger than 4σ . Equation (3) expresses the probability of a malicious client's ID falling in this range. $C(n, m)$ represents all combinations while selecting n numbers from a set of m numbers.

$$P_m = \frac{C(8\sigma, 1)C(D - 8\sigma, n_m - 1)}{C(D, n_m)} \quad (3)$$

Because $D \gg 8\sigma, n_m$, we can simplify these equations for the ease of analysis.

$$P_m = \frac{8\sigma n_m}{D} \quad (4)$$

We can also obtain the upper bound of the number of malicious clients under a given P_m . This upper bound gives the design rule for our collaborative FL framework.

$$n_m = \frac{DP_m}{8\sigma} \quad (5)$$

For instance, if the number of malicious clients is less than 2097, the bit length of D is 32 bits and $\sigma = 256$, then we can ensure that our FL architecture can resist poisoned attacks with P_m lower than 0.001. Hence, the probability that the number of the malicious clients' IDs exceeding 1 is close to 0. Hence, we assume the probability of having one malicious client misclassified into the normal group is low enough to ignore if we have well-designed parameters, D and σ .

The Impact of Prediction Accuracy of Global Models

Here, we try to show the impact of the prediction accuracy of the global model in a group on our proposed grouping scheme. In essence, regardless of the FL mechanisms, each global model in a group has its prediction accuracy for the input. Here, p_n denotes the prediction accuracy of the global model in a normal group while p_m denotes it in a malicious group. In a collaborative FL framework, the voting result decides the prediction output. Therefore, ensuring the correctness of the voting result is the main object of this collaborative FL framework. In order to achieve the object of this collaborative FL framework, the number of positive answers should be larger than that of negative ones. A positive answer means the true answer for the input test and a negative one means a false answer. Therefore, the expected number of positive answers is shown in Equation (6), and the expected number of negative answers follows accordingly.

$$N_{pe} = p_n N_n + p_m N_m \quad (6)$$

$$N_{ne} = N - N_{pe} \quad (7)$$

Moreover, we expect that $N_{pe} > N_{ne}$, and we assume that $N_n = N/2 + N_o'$ and $N_m = N/2 - N_o'$, where N_o' is the offset between the numbers of normal and malicious groups. Accordingly, we can deduce the minimum value for N_o' .

$$N_o' \geq \frac{N(1 - p_n - p_m)}{2(p_n - p_m)} \quad (8)$$

Usually, the malicious global models are useless for predicting the correct output results, which means these models predict outputs incorrectly. Hence, we set p_m to 0. Then,

$$N_o' \geq \frac{N(1 - p_n)}{2p_n} \quad (9)$$

However, to ensure that the number of normal groups is much larger than that of malicious groups, we stipulate that the number of normal groups should be $4N_o'$ more than the number of malicious ones. So, we set $N_o = 2N_o'$. N_n is at least $N_n = N/2 + N_o$ and N_m is at most $N/2 - N_o$. Hence, in our grouping scheme, we can obtain the minimum of N_n and the maximum of N_m in advance. In summary, we discover that a false positive rate of the edge clients can usually be ignored and prediction accuracy of the global model dominates the design of the number of normal and malicious groups.

3.5.3. Grouping Edge Clients

Next, we explain the process of the grouping scheme for all participating clients. In this study, we let the total groups be determined in advance, e.g., N . However, since different group categories have different values of ' k ', this variation can easily attract the attention of attackers. In order to prevent the adversary from discovering the grouping rule, k s in both group types are as similar as possible.

Hence, all groups use the same k . Before computing k , we should classify edge clients via the *distinguishing principle* from which we have n_{n_est} and n_{m_est} . In the context of the discussion about a false positive rate of classifying edge clients, n_{n_est} and n_{m_est} are almost equal to n_n and n_m . Thus, we can compute k , N_n and N_m in two cases as follows.

Case 1: in this case, $n_{n_est} \geq n_{m_est}$, k is equal to $\lceil n/N \rceil$ and all clients are assigned to their corresponding group. However, kN is usually higher than n , which means that the actual total number of groups is less than N . For the sake of ensuring total number of groups almost equals to N , some assigned normal edge clients should be assigned repeatedly into another normal group. The number of these repeatedly assigned normal edge clients is $kN - n$. Hence, the total number of normal edge clients needed to be assigned into groups is equal to $kN - n + n_{n_est}$. Therefore, N_n is equal to $\lceil (kN - n + n_{n_est})/k \rceil$ and N_m is equal to $\lceil n_{m_est}/k \rceil$. Some assigned malicious edge clients are assigned repeatedly until N_m are filled with malicious edge clients. In summary, based on our discussion above, we can obtain the following equation.

$$N_n = \max\left(\frac{N}{2} + N_o, \lceil (kN - n + n_{n_est})/k \rceil\right) \quad (10)$$

If N_n is $N/2 + N_o$, N_m should be $N/2 - N_o$. And if N_n is $\lceil (kN - n + n_{n_est})/k \rceil$, N_m is $\lceil n_{m_est}/k \rceil$. In the latter case, the maximum of $N_n + N_m$ may be equal to $N + 1$.

Case 2: in this case, $n_{n_est} < n_{m_est}$, we directly set N_n to $N/2 + N_o$ and N_m to $N/2 - N_o$. Then k is computed via $k = \lceil n_m/N_{m_est} \rceil$. The total number of malicious edge clients, kN_m , are assigned to malicious groups, where some malicious clients may be assigned repeatedly. Some normal edge clients are also assigned to normal groups repeatedly until all N_n are filled with normal edge clients.

Basically, we only consider k to be larger than 2. In our grouping scheme, only the malicious client just be assigned to malicious groups. Based on this rule, the malicious clients cannot influence normal groups. The following shows the pseudo code of our grouping scheme (Algorithm 1).

Algorithm 1 Grouping scheme

```

Input:  $C, N, \lambda, 4\sigma, p_n, p_m$ 
Output: Groups,  $N_n$ , and  $N_m$ ,
Initialization:  $C_n, C_m, N_o$ 
/* Classifying the edge clients and Counting the number of  $n_{n\_est}, n_{m\_est}$  */
For  $C_i$  in  $C$  do
    If the  $C_i$ 's IDs,  $Id_{n\_j}$ , is in the range of  $\lambda \pm 4\sigma$  Then
         $C_i$  is classified into the set of  $C_n$ 
    Else
         $C_i$  is classified into the set of  $C_m$ 
    End if
End for
Obtain  $n_{n\_est}, n_{m\_est}$  through the distinguishing principle
/* Computing  $k, N_n$ , and  $N_m$  */
If  $n_{n\_est} \geq n_{m\_est}$  Then
     $k$  is equal to  $\lceil n/N \rceil$ 
     $N_n = \max(\frac{N}{2} + N_o, \lceil (kN - n + n_{n\_est})/k \rceil)$ 
     $N_o = \frac{N(1-p_n-p_m)}{(p_n-p_m)}$ 
    If  $N_n = N/2 + N_o$ 
         $N_m = N/2 - N_o$ 
    If  $N_n = \lceil (kN - n + n_{n\_est})/k \rceil$ 
         $N_m = \lceil n_{m\_est}/k \rceil$ 
Else if  $n_{n\_est} < n_{m\_est}$  Then
     $N_n = N/2 + N_o, N_m = N/2 - N_o$ 
     $k = \lceil n_m/N_{m\_est} \rceil$ 
End if
/*Grouping all clients*/
While at least one of the normal groups is not filled with normal clients
    For  $C_i$  in  $C$  do
        If  $C_i \ni C_n$  Then
             $C_i \rightarrow N_{n\_j}$ , until this group is filled
            Then continue assigning  $C_i$  to next group, until all groups are filled with normal edge clients
        Else if  $C_i \ni C_m$  Then
             $C_i \rightarrow N_{m\_j}$ , until this group is filled
            Then continue assigning  $C_i$  to next group, until all groups are filled with malicious edge clients
        End if
    End for
End while

```

3.6. Dynamic Joining of Nodes

In our framework, new nodes that join the network post the initial setup phase will undergo a similar registration and ID assignment process as the initial nodes. These late entrants are assigned IDs by the central server, which are then used to integrate them into the existing collaborative FL framework. After ID assignment, these new nodes are categorized into groups based on the ID-distribution feature. This step is crucial as it ensures that the late entrants are appropriately integrated into existing normal or potentially malicious groups, depending on their assigned ID characteristics. If the existing group is full, the server creates a new group and the first member joining this group is regarded as the leader node. Once integrated into the groups, these late-joining nodes participate in the FL process just like the initial nodes. They contribute to their group's local model training, and consequently, to the global model development. Hence, our approach is also both feasible and scalable in large-scale environments. Both the central server's role in ID assignment and the dynamic nature of the grouping algorithm allow for flexible adaptation to the changing network topology. This ensures the framework's applicability in real-world, dynamic IoT and MUM-T scenarios.

4. Numerical Experiments

In this section, we conduct four numerical experiments to demonstrate that our grouping scheme effectively resists poisoning attacks in various scenarios. We summarize the purposes of these experiments as follows:

- The first numerical experiment demonstrates that our collaborative FL framework ensures the number of malicious groups remains less than that of normal ones, even if the number of malicious groups increases.
- The second experiment verifies that the overall prediction accuracy of our collaborative FL framework maintains a high value compared to FLcert-P, despite an increase in the number of malicious groups.
- The third experiment further proves that our collaborative FL framework performs well, regardless of the prediction accuracies of the malicious groups' global models.
- The final experiment confirms that our proposed collaborative FL framework maintains high prediction accuracy even if the prediction accuracies of some normal groups' global models are low.

The details of the experiment platform are listed in Table 1.

Table 1. Experiment platform.

Items	Specifications
Operation system	Windows 10 (Microsoft, Redmond, DC, USA)
Programming language	C++
CPU	11th Gen Intel(R) Core(TM) i9-11900F @ 2.50 GHz (Intel, Santa Clara, CA, USA)
Memory	32 G
Graphics card	Nvidia Geforce RTX 2060 (NVIDIA, Santa Clara, CA, USA)

4.1. Numerical Experiment 1

In the first numerical experiment, we set N to 50, n to 200, p_n to about 0.91 and p_m to 0. Hence, N_o is equal 5. Basically, this scenario represents a small-scale factory containing a few edge clients. We want to show our scheme can ensure that the number of normal groups must be larger than that of malicious groups in this scenario. Table 2 shows our analysis results.

Table 2. Numerical analysis.

n_m	N_m	FLcert-P (Worst Case)	Proposed Scheme
30		30	8
40		40	10
50		50	13
60		50	15
70		50	18
90		50	20
110		50	20
150		50	20

In this numerical experiment, FLcert-P randomly samples k clients to form a group based on its grouping rule, while our proposed grouping scheme aims to cluster all malicious clients into specific malicious groups. In FLcert-P, Nk/n is equal to 1, which means that all clients should be assigned to one group at most. To investigate a change in the

number of malicious groups in a collaborative FL framework using FLCer-P and our grouping scheme, this numerical experiment computes the number of malicious groups as the number of malicious edge clients increases. First, we present the number of malicious groups in FLCert-P under the worst case in which all malicious clients are assigned to completely different groups. Then, in our grouping scheme, the malicious clients are assigned to the malicious groups as much as possible. So, in contrast, our grouping scheme strives to assign malicious clients to malicious groups as much as possible. As observed from Table 2, our grouping scheme ensures that the number of malicious groups increases only up to $N/2 - N_o$ even as n_m continues to grow. Consequently, our proposed grouping scheme ensures that the number of normal groups will always exceed that of malicious groups. This indicates that our collaborative FL framework is likely to achieve higher predict accuracy compared to FLCert-P. The subsequent experiment is designed to prove this result.

4.2. Numerical Experiment 2

Next, in this experiment, we assume that p_m is nearly equal to zero. This implies that if an FL algorithm in a group is compromised by poisoning attacks, the prediction accuracy of its global model will become low, which leads to predicting outputs incorrectly. A notable example of such a federated algorithm is FedAvg, which is vulnerable to poisoning attacks. As a result, the prediction accuracy of a global model derived from FedAvg can drop to 0 even with a single attack. For a normal group, the prediction accuracy of its global model is set to p_n . In this experiment, we aim to demonstrate the output prediction accuracies of the overall collaborative FL framework for both FLCert-P and our grouping scheme, as the number of malicious edge clients increases.

To facilitate comparison with FLCert-P, we adopt accuracy as the evaluation metric. This metric represents the fraction of inputs correctly classified by the FL algorithm when m malicious clients are involved. We set N to 100, n to 300, and p_n to a random range between 0.92~0.95. According to the first experiment, it was revealed that the number of normal groups must exceed that of malicious ones in our framework. For FLCert-P, since $\binom{n}{k}$ is too large, we sample N groups according to our grouping rule, with each group containing k clients sampled uniformly at random from the n clients. As shown in Figure 3, the accuracy of our collaborative FL surpasses that of FLCert-P. When the number of malicious edge clients approaches 55, the prediction accuracy of FLCert-P decreases dramatically and is close to 0 while n_m approaches 75.

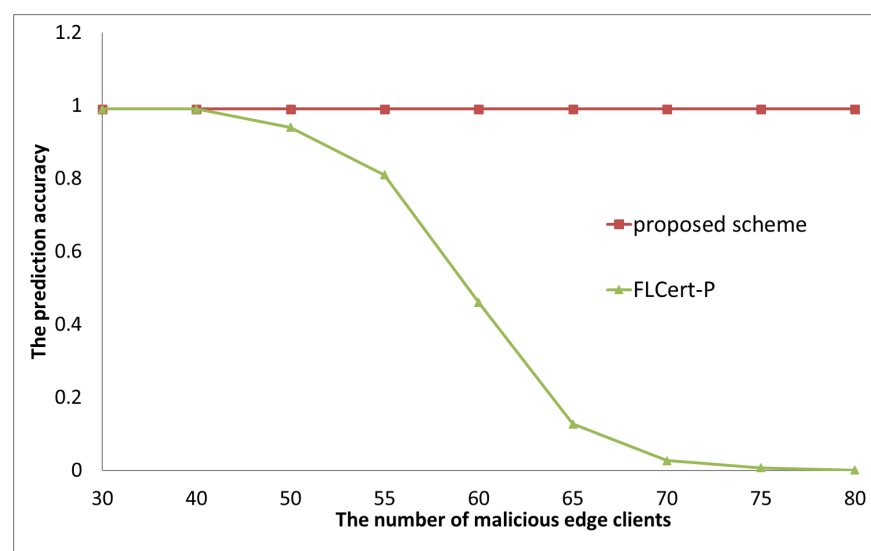


Figure 3. The accuracy of a collaborative FL as n_m increases.

4.3. Numerical Experiment 3

In this experiment, we evaluate the impact of the attack ability of poisoning attacks on the accuracy of a collaborative FL framework. Here, we use p_m to represent the attack ability, as the poisoning attack's primary aim is to decrease the prediction accuracy of the malicious groups. This numerical experiment assesses the resilience of our proposed collaborative FL framework under various values for p_m . In this experiment, we set N to 100, n to 300, and p_n to a randomly selected range between 0.92~0.95.

We consider three scenarios: $p_m = 0, 0.4$, and 0.6 . As illustrated in Figure 4, our proposed collaborative FL framework performs well across various p_m values even when many malicious edge clients are present in the network. As for FLcert-P, its accuracy is highly dependent on the FL algorithm's ability to resist poisoning attacks. Therefore, if an FL algorithm effectively counters poisoning attacks, maintaining a high p_m , then FLcert-P is likely to perform well. However, developing such an ideal FL algorithm requires significant effort and still remains a future goal.

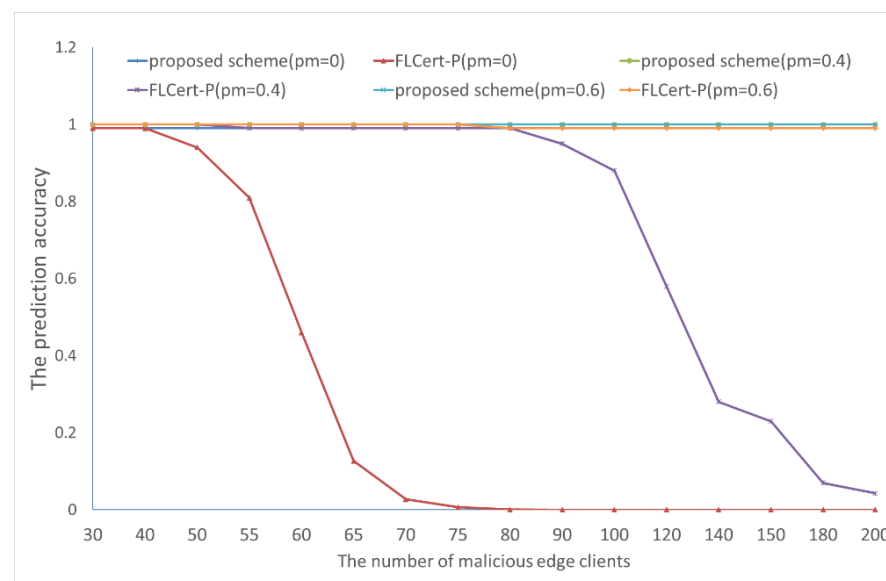


Figure 4. The accuracy of a collaborative FL for different p_m .

4.4. Numerical Experiment 4

Basically, for an FL framework, the global model may not achieve sufficiently high prediction accuracy for edge clients. In our study, we use the parameter p_n to express the prediction accuracy of one normal group. Now, in this experiment, we want to show that our proposed collaborative FL framework can still deliver high enough prediction accuracy even if certain normal groups' global models have low prediction accuracies. That is because our grouping scheme can use N_o to compensate for the lack of high accuracy of a global model. Hence, we set p_n to approximately 0.75~0.95 for all normal global models and p_m to 0 for malicious global models. In this experiment, we also set N to 100, and n to 300. The results, as depicted in Figure 5, indicate that our proposed collaborative FL framework retains high performance. Conversely, Flcert-P is shown to not tackle this situation well, as is unfortunately often found to be the case in a collaborative FL framework.

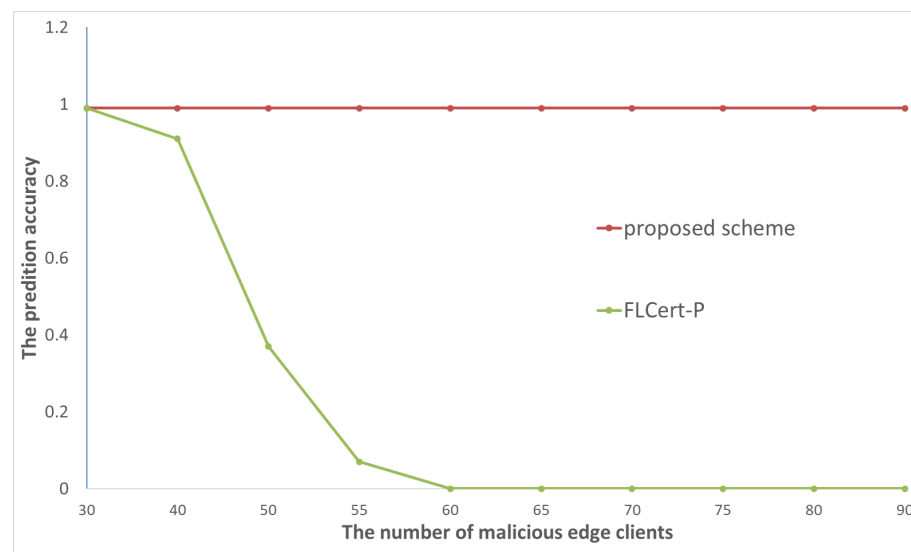


Figure 5. The accuracy of a collaborative FL framework for low p_n .

5. Conclusions

In this study, we developed a novel grouping scheme in a collaborative FL framework for defending against poisoning attacks. Our scheme, leveraging ID-distribution features, effectively manages the categories of participating groups and counters poisoning attacks. We established a boundary for the maximum number of malicious groups, ensuring they are outnumbered by normal groups. This strategic approach with a voting game principle, enhances the accuracy and reliability of the FL process. Through several numerical experiments, our framework demonstrated robust resistance to model poisoning attacks, consistently maintaining high accuracy across varying scenarios. The results affirm that our method effectively ensures a higher number of normal groups, thus enhancing the overall prediction accuracy and robustness of the FL framework against security threats.

Author Contributions: Formal analysis, C.-H.C. and C.-F.L.; investigation, C.-H.C., Z.-Y.S., I.-H.L. and C.-K.L.; validation, Z.-Y.S. and C.-F.L.; methodology, Z.-Y.S. and C.-K.L.; project administration, C.-H.C. and C.-K.L.; writing—original draft, C.-H.C. and Z.-Y.S.; writing—review and editing, C.-K.L. and I.-H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Science and Technology Council (NSTC) in Taiwan under contract number NSTC 112-2634-F-006-001-MBK.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to technical limitation.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

Symbols	Definitions
C	The set of the edge clients
$C_{n,i}$	The i th normal edge client
C_i	The i th edge client
$C_{m,j}$	The j th malicious client
N	The total number of groups
N_m	The number of malicious groups
$N_{m,j}$	The j th malicious group

N_n	The number of normal groups
$N_{n,j}$	The j th normal group
n	The number of total clients
n_n	The number of normal clients
n_m	The number of malicious clients
n_f	The number of malicious clients whose IDs fall in valid range
k_n	The number of client members in a normal group
k_m	The number of client members in a malicious group
k	The number of client member in a group
N_{pe}	The expected number of positive answers
N_{ne}	The expected number of negative answers
p_n	The prediction accuracy of the global model in a normal group
p_m	The prediction accuracy of the global model in a malicious group
N_o	The number offset between normal and malicious groups
λ	The mean of normal distribution
σ	The standard deviation of normal distribution

References

1. Kone, J.; McMahan, H.B.; Yu, X.F.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. In Proceedings of the NeurIPS Workshop Private Multi-Party Machine Learning 2016, Barcelona, Spain, 9 December 2016. [\[CrossRef\]](#)
2. Qolomany, B.; Ahmad, K.; Al-Fuqaha, A.; Qadir, J. Particle Swarm Optimized Federated Learning for Industrial IoT and Smart City Services. In Proceedings of the GLOBECOM 2020–2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020. [\[CrossRef\]](#)
3. Xing, J.; Jiang, Z.X.; Yin, H. Jupiter: A Modern Federated Learning Platform for Regional Medical Care. In Proceedings of the 2020 IEEE International Conference on Joint Cloud Computing, Oxford, UK, 3–6 August 2020. [\[CrossRef\]](#)
4. Hu, Y.; Cao, N.; Guo, W.; Chen, M.; Rong, Y.; Lu, H. FedDeep: A Federated Deep Learning Network for Edge Assisted Multi-Urban PM2.5 Forecasting. *Appl. Sci.* **2024**, *14*, 1799. [\[CrossRef\]](#)
5. Saputra, Y.M.; Hoang, D.T.; Nguyen, D.N.; Dutkiewicz, E.; Mueck, M.D.; Srikanteswara, S. Energy Demand Prediction with Federated Learning for Electric Vehicle Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019. [\[CrossRef\]](#)
6. Yu, Z.; Hu, J.; Min, G.; Zhao, Z.; Miao, W.; Hossain, M.S. Mobility-Aware Proactive Edge Caching for Connected Vehicles Using Federated Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5341. [\[CrossRef\]](#)
7. Li, H.; Li, C.; Wang, J.; Yang, A.; Ma, Z.; Zhang, Z.; Hua, D. Review on security of federated learning and its application in healthcare. *Future Gener. Comput. Syst.* **2023**, *144*, 271–290. [\[CrossRef\]](#)
8. Rahman, S.A.; Tout, H.; Talhi, C.; Mourad, A. Internet of Things Intrusion Detection: Centralized, on-Device, or Federated Learning? *IEEE Netw.* **2020**, *34*, 310. [\[CrossRef\]](#)
9. Cao, X.; Zhang, Z.; Jia, J.; Gong, N.Z. FLCert: Provably Secure Federated Learning Against Poisoning Attacks. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 3691. [\[CrossRef\]](#)
10. Liu, C.K.; Chiang, C.H. A Collaboration Federated Learning Framework with a Grouping Scheme against Poisoning Attacks. In Proceedings of the International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 30 June–3 July 2023. [\[CrossRef\]](#)
11. Ghimire, B.; Rawat, D.B. Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things. *IEEE Internet Thing J.* **2022**, *9*, 8229. [\[CrossRef\]](#)
12. Taheri, R.; Shojafar, M.; Alazab, M.; Tafazolli, R. Fed-IIoT: A Robust Federated Malware Detection Architecture in Industrial IoT. *IEEE Trans. Ind. Informat.* **2021**, *17*, 8442. [\[CrossRef\]](#)
13. Sun, Y.; Ochiai, H.; Esaki, H. Intrusion Detection with Segmented Federated Learning for Large-Scale Multiple LANs. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020. [\[CrossRef\]](#)
14. Lin, K.-Y.; Huang, W.-R. Using Federated Learning on Malware Classification. In Proceedings of the 22nd International Conference on Advanced Communication Technology, Phoenix Park, Republic of Korea, 16–19 February 2020. [\[CrossRef\]](#)
15. Fu, A.; Zhang, X.; Xiong, N.; Gao, Y.; Wang, H.; Zhang, J. VFL: A Verifiable Federated Learning with Privacy-Preserving for Big Data in Industrial IoT. *IEEE Trans. Ind. Informat.* **2022**, *18*, 3316. [\[CrossRef\]](#)
16. Ioannou, I.; Nagaradjane, P.; Angin, P.; Balasubramanian, P.; Kavitha, K.J.; Murugan, P.; Vassiliou, V. GEMILDS-MIOT: A Green Effective Machine Learning Intrusion Detection System based on Federated Learning for Medical IoT network security hardening. *Comput. Commun.* **2024**, *218*, 209–239. [\[CrossRef\]](#)
17. Khoa, T.V.; Saputra, Y.M.; Hoang, D.T.; Trung, N.L.; Nguyen, D.; Ha, N.V.; Dutkiewicz, E. Collaborative Learning Model for Cyberattack Detection Systems in IoT Industry 4.0. In Proceedings of the IEEE Wireless Communications and Networking, Seoul, Republic of Korea, 25–28 May 2020. [\[CrossRef\]](#)

18. Huong, T.T.; Bac, T.P.; Long, D.M.; Luong, T.D.; Dan, N.M.; Quang, L.A.; Cong, L.T.; Thang, B.D.; Tran, K.P. Detecting cyberattacks using anomaly detection in industrial control systems: A Federated Learning approach. *Comput. Ind.* **2021**, *132*, 103509. [[CrossRef](#)]
19. Mahindru, A.; Arora, H. DNNdroid: Android Malware Detection Framework Based on Federated Learning and Edge Computing. In Proceedings of the Advancements in Smart Computing and Information Security 2022, Rajkot, India, 24–26 November 2022. [[CrossRef](#)]
20. Mahindru, A.; Sharma, S.K.; Mittal, M. YarowskyDroid: Semi-supervised based Android malware detection using federation learning. In Proceedings of the 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 5–6 May 2023; pp. 380–385. [[CrossRef](#)]
21. Gálvez, R.; Moonsamy, V.; Diaz, C. Less is More: A privacy-respecting Android malware classifier using Federated Learning. *Proc. Priv. Enhancing Technol.* **2021**, *4*, 96–116. [[CrossRef](#)]
22. Jiang, C.; Yin, K.; Xia, C.; Huang, W. FedHGCDroid: An Adaptive Multi-Dimensional Federated Learning for Privacy-Preserving Android Malware Classification. *Entropy* **2022**, *24*, 919. [[CrossRef](#)] [[PubMed](#)]
23. Lamport, L.; Shostak, R.; Pease, M. The Byzantine Generals Problem. *ACM Trans. Prog. Lang. Sys.* **1982**, *4*, 382. [[CrossRef](#)]
24. Blanchard, P.; Mhamdi, E.M.E.; Guerraoui, R.; Stainer, J. Machine learning with adversaries: Byzantine tolerant gradient descent. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017. [[CrossRef](#)]
25. Zhang, Z.; Cao, X.; Jia, J.; Gong, N.Z. FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 14–18 August 2022. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.