SCA: Sybil-based Collusion Attacks of IIoT Data Poisoning in Federated Learning

Xiong Xiao, Zhuo Tang, Chuanying Li, Bin Xiao, and Kenli Li

***Abstract*—With the massive amounts of data generated by Industrial Internet of Things (IIoT) devices at all moments, federated learning (FL) enables these distributed distrusted devices to collaborate to build machine learning model while maintaining data privacy. However, malicious participants still launch malicious attacks against the security vulnerabilities during model aggregation. This paper is the first to propose sybil-based collusion attacks (SCA) in the IIoT-FL system for the vulnerabilities mentioned above. The malicious participants use label flipping attacks to complete local poisoning training. Meanwhile, they can virtualize multiple sybil nodes to make the local poisoning models aggregated with the greatest possibility during aggregation. They focus on making the joint model misclassify the selected attack class samples during the testing phase, while other non-attack classes kept the main task accuracy similar to the non-poisoned state. Exhaustive experimental analysis demonstrates that our SCA has superior performance on multiple aspects than the state-of-the-art.**

***Index Terms*—IIoT, Federated learning, Label flipping attacks, Sybil, Collusion attacks.**

I. Introduction

WITH the fast development of industry 4.0 and the widespread popularity of industrial Internet of Things (IIoT) applications makes applications such as smart transportation and smart healthcare thrive and also makes the data generated by the industrial devices exponentially grow. Such as autonomous driving technology [1], it needs to train all data generated by sensor and camera devices to build a stable joint model to identify road conditions. And the distributed IIoT devices can generate a large amount of data in a short time [2]. In order to take into account the efficiency of processing big data and protect the privacy of clients. A novel machine learning paradigm named federated learning (FL) [3] was proposed, which is a new solution based on distributed training to alleviate the performance bottleneck and privacy risk caused by centralized processing. Traditional machine learning methods [4] usually store and run these data centrally, which will generate considerable computational and communication overhead in involving millions of mobile devices or massive data. This makes it unacceptable for sensitive IIoT applications (e.g., autonomous driving, intelligent robots, smart medical) that require real-time data transmission [5]. In addition, relying on centralized storage will cause a huge risk of private leakage [6]. Generally, when FL performs the collaborative training process of multiple distributed participants (e.g., IIoT devices), the sensitive information and private data of each client are kept locally [7]. FL has demonstrated excellent performance in the distributed execution process, while ensuring the privacy of participants by performing independent local training and model updates, so as to implement collaborative calculating in a joint environment that includes malicious participants. This also makes FL attract much attention in many fields including smart healthcare [8] [9], smart feature prediction [10], and Internet of Things in smart homes [11] [12].

The IIoT represents a distributed network composed of intelligent and highly interconnected industrial devices, each device can act as an FL participant to participate in training and updating [13]. FL improves the performance of the model for IIoT applications through continuous iterative training, and finally obtains a stable global model when the iteration reaches convergence. However, FL greatly exposes its weaknesses to malicious adversaries during the process of performing training [14]. Malicious adversaries can obtain the information of the global model in each round and upload malicious parameters or perform a small part of the beneficial contribution for collaborative training while avoiding anomaly detection as much as possible. For instance, malicious adversaries use contaminated data for training locally [15] [16], or tamper and prune local models for poisoning aggregation [17] [18].

The existing works [12] [19] have shown that controlling more malicious IIoT devices or using more direct poisoning attacks during the execution of FL is more destructive to the global model. Due to network, communication, power, and other issues in a heterogeneous federated environment, many IIoT devices are at risk of offline. Malicious participants will virtualize multiple malicious nodes in this unstable communication network. With more significant damage to the construction of the shared global model, this byzantine fault-tolerance [20] problem usually uses the technology of fusion sybil-based attacks. In addition, in the process of malicious participants performing poisoning attacks, they usually use mislabeled samples for training or upload the poisoned models to the central server for aggregation. Compared with the independent attacks by a single malicious participant, the

The work is supported by the National Key Research and Development Program of China (2018YFB1701400), the National Natural Science Foundation of China (Grant Nos. 92055213, 61873090, L1924056), Guangdong Province Research and Development Plan Project in Key Area (Grant Nos. 2020B0101100001), China Knowledge Centre for Engineering Sciences and Technology Project (CKCEST-2021-2-7), Shenzhen Development and Reform Commission Strategic Emerging Industry Development Project (XMHT20190205007).

Xiong Xiao, Zhuo Tang, Chuanying Li and Kenli Li are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China, and also with the National Supercomputing Center in Changsha, Hunan University, Changsha 410082, China. Xiong Xiao and Bin Xiao are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: xx@hnu.edu.cn, ztang@hnu.edu.cn, lichuanying@hnu.edu.cn, b.xiao@polyu.edu.hk, lkl@hnu.edu.cn.

Zhuo Tang and Bin Xiao are the author for correspondence.

collusion attacks by multiple malicious participants have a higher attack success rate and can better obscure their attack behavior. Meanwhile, due to the characteristics of data privacy protection, the central server cannot verify the local data of all participants, and the parameter transmission process of all participants is anonymous, which provides more possibilities for malicious participants to launch malicious attacks.

Therefore, in order to better focus on the implementation of poisoning attacks in the IIoT-FL system, in this work, we introduce an efficient sybil-based collusion attacks (SCA) scheme. We represent the malicious IIoT device as a malicious participant in our system. Precisely, first, in the FL computing environment that we set, all the participants can only control local data and cannot access the data of other participants. This enables them to better manipulate local data for poisoning training without being detected. The most commonly used data poisoning methods are backdoor poisoning attacks [15] and label flipping attacks [16]. This paper uses label flipping attacks to conduct poisoning training on the massive data generated by IIoT devices, aiming to make the global model misclassify the selected attack class samples. However, the attack effect achieved by such an attack is insufficient. Second, we use the cloning properties of sybil that all sybil nodes virtualized by malicious participants will perform the same malicious operations during the training process and have equal attack influence. We consider making the malicious model has a higher probability to be aggregated during FL aggregation. Finally, we collude with all malicious participants to launch the collusion attacks, aiming to replace the global model using the poisoning model. Meanwhile, such collusion attacks can better obscure their attack behavior. We utilize Fashion MNIST and CIFAR-10 datasets to represent the data generated by IIoT devices and conduct experiments. In summary, our contributions to this work are mainly in four-fold as below.

- We explore sybil-based collusion attacks of IIoT data poisoning for the IIoT-FL application, and implement poisoning training and model collusion attacks in this IIoT-FL system.
- We make minimal malicious assumptions for malicious adversaries and integrate the label flipping poisoning attacks to make the global model misclassify the selected attack class samples while maintaining the main task accuracy of other non-attack classes.
- We further propose an efficient sybil-based collusion attacks (SCA) method, which aims to make the poisoning collusion models to be aggregated with greater probability during aggregation, and successfully obscure their attack behavior.
- We utilize F-MNIST and CIFAR-10 datasets to represent the data generated by IIoT devices. Exhaustive experimental analysis demonstrates that our SCA has superior performance than the state-of-the-art.

The remainder of this paper is organized below. Section II primarily describes the background knowledge and corresponding comments of the relevant literature. Section III mainly focuses on introducing the FL problem formulation and threat model. Section IV discusses the proposed SCA based on the IIoT-FL computing environment. Section V analysis the performance results of our proposed SCA in multiple attack scenarios. Section VI summarizes the full paper.

II. Related Work

A. Poisoning Attacks in Federated Learning System

In distributed training for large-scale IIoT devices, FL has demonstrated excellent performance while maintaining data privacy. In the FL system, although all participants do not share data and will not disclose local data to the central server simultaneously, such a system still has a huge security risk during the training process. Malicious participants manipulate the local poisoning data for training or tamper with the local model for poisoning aggregation, which will undoubtedly make the performance of the global model have a significant impact. This kind of poisoning attack will also make some IIoT applications cause fatal security accidents in actual scenarios [21]. Specifically, it is divided into the following two attack scenarios: *Model poisoning attacks:* It occurs in the local model update stage. Malicious adversaries update the poisoned local model to the central server for aggregation. The attack effect is global, and the attack target is arbitrary. *Data poisoning attacks:* It usually attacks the local data of edge devices. The malicious adversary uses the poisoned data for training, and this type of attack is specific to the attack class. The data-based attacks include two methods, label flipping poisoning attacks and backdoor poisoning attacks. Xie *et al.* [22] manipulated a subset of training data by injecting adversarial triggers to perform the wrong prediction on images embedded with triggers in a distributed heterogeneous dataset. Sun *et al.* [23] injected backdoor tasks into a part of the images to damage the global model's performance on the target task. Although it has a high attack success rate, it can cause much overhead to inject backdoor triggers into large-scale training samples. In addition, the goal of our attack is to misclassify the selected attack class samples. So in this work, we use the label flipping poisoning attacks. Malicious adversaries can perform label flipping attacks without conducting parameter interaction, changing the FL architecture, and pre-training. They use the dirty data with the wrong label for training locally. This attack method is both concealed and direct.

B. Sybil-based Attacks on Federated Learning Model

In a federated learning system involving a large number of mobile devices or IIoT sensors, these devices are usually disconnected or inactivated due to network (e.g., 5G, 4G, WIFI), communication delay, power, and other issues. A computing system that supports such participants to leave or join intermittently is vulnerable to sybil-based attacks [24]. Generally speaking, sybil attack nodes will create multiple malicious identities by forging or compromising other honest IIoT devices. They focused on malicious attacks to damage the federated learning model's performance by using false identities under multiple aliases to enhance their attack influence. All false nodes will perform the same malicious operations and have an equal attack impact. Jiang *et al.* [25] proposed a sybil-based attacks method. Sybil clients compromised the infected

device to update the poisoning model directly. They proved their effectiveness on several advanced defense methods, while also slowing down the convergence of the global model. Fung *et al.* [26] also designed a novel sybil-based attacks technology, it has shown the effectiveness on multiple recent distributed machine learning fault tolerance protocols. The sybil attacks also showed an excellent attack effect in IoT applications [27]. Although they have shown reliability in the attack effect, the drift gradient of their local poisoning model is very easy to detect and remove. In this paper, we integrate the sybil-based collusion attacks technology to make the local poisoning model have a higher possibility of aggregation and help malicious participants better obscure the attack behavior.

C. Collusion Attacks for IIoT in Federated Learning

When training a global model of IIoT application, multiple malicious adversaries can collude to launch joint attacks. They uploaded the collude malicious parameters to the server for aggregation simultaneously, and performed iterative attacks to destroy the performance of the model. Taheri *et al.* [28] proposed two dynamic poisoning attack strategies that integrate Generative Adversarial Network (GAN) and Federated Generative Adversarial Network (FedGAN) on the side of the participants, and evaluated them on IIoT applications. Lim *et al.* [29] studied the collusion attacks between dishonest participants and the server. The malicious participant uploads the poisoning model during the aggregation stage, and the server also leaks the parameters of other participants to the malicious participant. They aim to achieve the purpose of reducing the global model's performance while analyzing the local model of other participants to avoid anomaly detection [30] during the poisoning process. However, this method needs to make more malicious assumptions about the federated system. Meanwhile, it also generates a lot of communication overhead during the collusion phase between the participants and the server. In this paper, we just focus on collusion attacks between malicious participants. They only manipulate local data and control sybil nodes without destroying the overall architecture of federated learning. Such potentially malicious scenarios exist in most IIoT applications.

III. Preliminaries and Threat Model

A. IIoT Based on Federated Learning

The computing scenario that integrates IIoT and federated learning (IIoT-FL) supports distributed multiple participants (e.g., sensors, IIoT devices) for collaborative training. In general, such a pattern is set to *client-server*. As shown in Fig.1, each IIot device (C = $\{c_i\}_{i=1}^{n}$) uses their local data (D = $\{D_{c_i}\}_{i=1}^{n}$) to build a local machine learning model M_{Loc}. Then update the local model to the central server for aggregation (Agg) to train a new global model M_{Glo}. This distributed training process guarantees the same model quality as the centralized training method and maintains the data privacy of the participants. The four main execution steps of IIoT-FL include system initialization, local training, model update, and aggregation. When iterative training converges, a stable global model will be constructed. Specifically, the training data of each participant consists of multiple samples D_{c_i} = (s_1,s_2,...s_n). Each sample s_i is composed of a set of features f_i and corresponding class labels y_i, $y_i \in Y$, where Y is the set of classes of all labels in the dataset. When the sample is input, the output result obtained by the Softmax function is a set of predicted classification probabilities (pcp_y) corresponding to all classes. For each sample, the global model calculates $M_{Glo}(s_i) = argmax(pcp_y)$ for correct classification. Meanwhile, the cross-entropy loss function L is calculated to find the parameter value of the minimized loss through continuous iterative calculations. Each participant minimizes the local loss by executing the Stochastic Gradient Descent (SGD) algorithm in the local training phase. FL aims to train a global model that can correctly identify all test samples. Otherwise, when the autonomous vehicles recognize the stop sign as a walking sign, a serious traffic accident may occur [31]. In addition, in each round r of iteration, the server randomly selects a group of o participants P_o, and adopts the Federated Averaging (FedAvg) algorithm [3] to aggregate their local model M_{Loc}. The aggregation process is denoted as Eq.1

$$M_{Glo}^{(r)} = \frac{1}{o}\sum_{k=1}^{o} M_{Loc_k}^{(r)} \tag{1}$$

In the next round, the newly aggregated global model $M_{Glo}^{(r)}$ is distributed to all n participants in the FL system. After all the participants get the global model $M_{Glo}^{(r)}$ from the server, they execute the following SGD algorithm as Eq. 2 locally to build a new local model of $r+1$ round and upload it.

$$M_{Loc}^{(r+1)} = M_{Glo}^{(r)} - \eta \cdot \nabla L(M_{Glo}^{(r)}, D_{c_i}) \tag{2}$$

where ∇L and η represent gradient and learning rate, respectively. Until all R rounds are executed, the iterative training is completed, and the global model M_{Glo} is obtained.

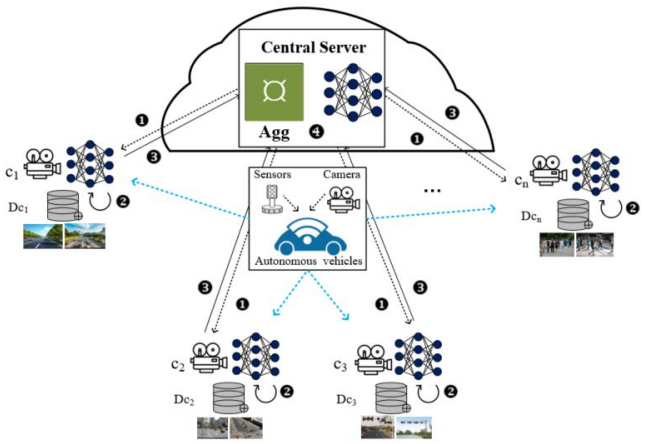


Fig. 1. The execution workflow and architecture of IIoT-FL system (e.g. autonomous driving system). The four main communication processes in current system have been highlighted.

B. Adversary Objective and Capabilities

Adversary model: We focus on studying sybil-based collusion attacks in IIoT-FL. As shown in Fig.2, n IIoT clients are

trained in collaboration in the current FL system. We highlight in black for honest participants and highlight in red for malicious participants. Moreover, each malicious participant can virtualize multiple sybil malicious nodes to participate in malicious attacks on the system. We consider that all malicious participants are the main adversaries of the system, and they use poisoned data for training. Among n IIoT clients, we set $k\%$ as the proportion of malicious adversaries. Therefore, the total number of malicious adversaries is represented by $K = n * k\%$. Meanwhile, each malicious adversary can virtualize v sybil nodes. $MA = K*v+K$ malicious nodes will participate in the system for aggregation. They have a higher probability of being aggregated, and the selected sybil node will eventually execute the attacks of the original malicious node. In addition, we assume that the server of the current IIoT-FL system is legitimate, and malicious adversaries will not damage the algorithm and network architecture.

Adversary objective: In our attack settings, we perform targeted collusion attacks. We consider the scenario that the malicious adversary uses the label flipping strategy to train the poisoning data locally and collude with other poisoning models. Meanwhile, the malicious adversary virtualizes multiple sybil nodes in the system, so that the server selects the collusion model to perform aggregation with greater probability, thereby constructing a global poisoning model. All malicious adversaries have the defined and same goal: to make the global model misclassify the source class samples while maintaining the classification accuracy of other main task classes. Furthermore, the collusion attacks by malicious participants aim to evade parameter anomaly detection more effectively.

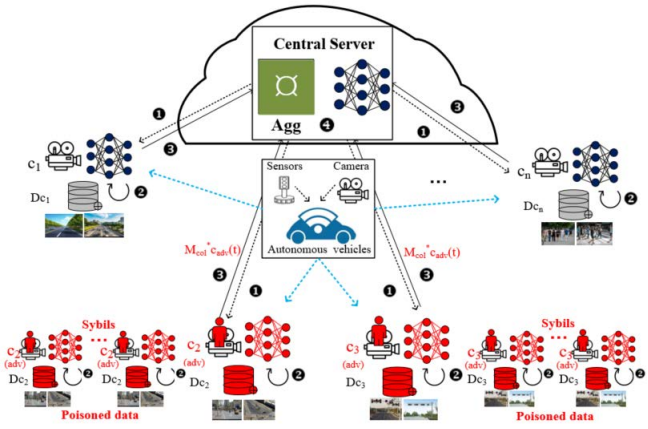


Fig. 2. Adversary model in IIoT-FL system, and n IIoT clients in current IIoT-FL system. We highlight the adversaries in red, and others in black.

Adversary capabilities: All participants in the federated environment aim to make the most outstanding contribution to improving the global model's performance, except the malicious participants. Malicious adversary must exert greater influence on the model than honest participant that can achieve an effective attack. In addition, we assume that the server in the FL system is legitimate, it will not collude with malicious adversaries to destroy the model, nor will it disclose the private information of all participants. This also gives malicious participants more space for local poisoning training. In this paper, the malicious adversary will not steal the data privacy of other participants, nor will it destroy the model structure and the integrity of aggregation algorithm. They have the ability to create sybil nodes and launch collusion attacks invisibly.

IV. Sybil-based Collusion Attacks Strategies

A. Label Flipping Poisoning Attacks in IIoT-FL System

The malicious adversary aims to make the IIoT-FL global model misclassify the attacked selected class without interfering with the normal classification of other classes. Therefore, using label flipping attacks is the best way. It can poison the entire selected class of the dataset without the need for additional pre-training. The malicious adversary can launch such an attack invisibly and achieve a more direct attack effect. All malicious participants use label flipping attacks locally to generate selected poisoned samples. Specifically, for a certain class of samples in the dataset, the malicious adversary only modifies its label without performing other operations (e.g., adding noise, marking features). For the selected source class label y_{src} and a given target class label y_{tar} from Y, the malicious adversary will replace the selected attack class label with the source class label to launch a data poisoning attacks. In order to show the effectiveness of our proposed attack strategy more clearly in the IIoT-FL system. We set two goals for the attack. We take the two groups of samples (containing source and target classes) that are the hardest and easiest to misclassify among all samples as our experimental baseline.

B. SCA Based on Label Flipping Poisoning Attacks

1) Sybil nodes virtualization: The computing system that supports participants to leave or join intermittently is vulnerable to sybil-based attacks. The poisoning model of malicious participants are cloned by their sybils, which is denoted as below.

$$M_{Loc_{sybi}}^{(r+1)} = M_{Glo}^{(r)} - \eta \cdot \nabla L(M_{Glo}^{(r)}, D_{c_{sybi}}) \tag{3}$$

where $M_{Loc_{sybi}}^{(r+1)}$ and $M_{Glo}^{(r)}$ represent the local sybil poisoning model of $r+1$ round and the global model of r round.

They aim to poison the global model during aggregation. Due to the characteristics of privacy protection in federated learning [14], the IIoT-FL system cannot verify the local data of all participants, nor will it detect their local training process, and the parameter transmission process of all participants is anonymous. Meanwhile, all the participants can only control local data and cannot access the data of other participants, which gives sybils more space for poisoning. Besides, more virtual sybils will have a considerable malicious impact on the overall system, and they have a greater possibility of aggregation. The updates of honest participants will be overwhelmed by the poisoned updates of malicious adversaries, thus making the global model develop in a malicious direction.

2) SCA on IIoT-FL model: All malicious adversaries (including original malicious participants and sybil nodes) get their local poisoning model by using Eq. 3. Although they can make the global model misclassify the samples of attack class.

Algorithm 1: SCA Algorithm in IIoT-FL System

Input: Initial global model $M_{Glo}^{(0)}$, Local training data (Involving poisoning data) D_{c_i}, Communication round r, Virtual sybil nodes v, Learning rate η, Loss function L, Epoch E and Batch size of local dataset b

Output: $M_{Glo}^{(r+1)}$

Initialize malicious participants K in C;
//Server executes AGGREGATE(r+1);
for $c_i \in P_o$ **do**
  $M_{Loc_{c_i}}^{(r+1)} = LOCALUPDATE(M_{Glo}^{(r)})$
end
$M_{Glo}^{(r+1)} = \frac{1}{o}\sum_{k=1}^{o} M_{Loc_{c_i}}^{(r+1)}$;
//Sybil virtualization from Malicious Participants;
Sybil nodes $= K * v$;
$C = C + K * v$;
malicious participants $\{c_{advi}\}_{i=1}^{K}$;
for $c_{advi} = (1...K)$ **do**
  for $Sybils\_i \in v * K$ **do**
    $M_{Loc_{sybi}}^{(r+1)} = M_{Glo}^{(r)} - \eta \cdot \nabla L(M_{Glo}^{(r)}, D_{c_{sybi}})$
  end
end
//Clients executes LOCALUPDATE($M_{Glo}^{(r)}$);
//Local Updates from Honest Participants;
honest participants $c_i = 1, c_i \in (C - MA)$;
for $epoch\_i = (1...E)$ **do**
  for $localbatch\_b \in D_{c_i}$ **do**
    $M_{Loc_{c_i}}^{(r+1)} = M_{Glo}^{(r)} - \eta \cdot \nabla L(M_{Glo}^{(r)}, b)$
  end
end
$M_{Loc_{c_i}}^{(r+1)} \leftarrow M_{Glo}^{(r)}$;
//Local Updates from Malicious Adversaries;
malicious adversaries $c_{advi} = 1, c_{advi} \in MA$;
$MA = K * v + K$;
for $epoch\_i = (1...E)$ **do**
  for $localbatch\_b \in D_{c_{advi}}$ **do**
    $M_{Loc_{advi}}^{(r+1)} = M_{Glo}^{(r)} - \eta \cdot \nabla L(M_{Glo}^{(r)}, b)$
  end
end
$M_{Loc_{adv}}^{(r+1)} = \frac{1}{MA}\sum_{i=1}^{MA} M_{Loc_{advi}}^{(r+1)}$;
for $c_{advi} = (1...MA)$ **do**
  $M_{Loc_{advi}}^{(r+1)} \leftarrow M_{Loc_{adv}}^{(r+1)}$;
end
$M_{Loc_{advi}}^{(r+1)} \leftarrow M_{Glo}^{(r)}$;
return $M_{Glo}^{(r+1)}$;

However, due to the differences in the local data trained by each malicious participant, the gradient of the model training has a high drift, so it is easy to be found by the parameter detection method (e.g., anomaly detection) [30]. In order to better obscure their attack behavior, we merge all local poisoning models to perform collusion attacks operations, update the gradient parameters after fine-tuning, and perform aggregation. The collusion attacks is denoted as Eq. 4

$$M_{Loc_{adv}}^{(r+1)} = \frac{1}{MA}\sum_{i=1}^{MA} M_{Loc_{advi}}^{(r+1)} \tag{4}$$

where $M_{Loc_{adv}}^{(r+1)}$ is the collusion model, and the adv includes all malicious participants and sybil nodes. Then the collusion model will be redistributed to all malicious adversaries. They have a greater probability of aggregation and a significant adverse effect on the global model. After executing SCA, if the server aggregates the poisoned local models, the aggregation process will be transformed as follows:

$$M_{Glo}^{(r+1)} = \frac{1}{o}(\sum_{k=1}^{o-u} M_{Loc_k}^{(r+1)} + \sum_{k=1}^{u} M_{Loc_{advk}}^{(r+1)}) \tag{5}$$

where u represents the number of aggregated malicious adversaries (including malicious participants and sybils). If all local models come from malicious adversaries, that is, when $o = u$, then the global model $M_{Glo}^{(r+1)}$ will be directly replaced by the malicious model M_{Mal}, which is represented as Eq. 6

$$M_{Mal} = \frac{1}{o}\sum_{k=1}^{o} M_{Loc_{advk}}^{(r+1)} \tag{6}$$

Algorithm 1 describes the execution process of SCA in detail. The entire IIoT-FL system is coordinated by n participants to train a global model, including multiple malicious adversaries, who use local poisoning data for training. Besides, they receive the initial model as other honest participants and use the same loss function as well as aggregation algorithm.

V. Experiments Analysis

A. Datasets, Model Architecture and Experiment Setup

Datasets: We verify the attack performance of our designed SCA to the image classification task and implement a federated learning prototype on two widely adopted benchmark datasets (Fashion-MNIST and CIFAR-10) to evaluate our method. F-MNIST [32] and CIFAR-10 [33] are used to represent the data generated by IIoT devices. The F-MNIST dataset includes 60,000 gray-scale images (28×28) for training and 10,000 images for testing. The CIFAR-10 dataset contains 50,000 color images (32×32) for training and 10,000 images for testing. Meanwhile, we study our attack performance on two data distributions, including iid and Non-iid. For iid, we shuffle the data and divide it into all the participants. For Non-iid, where the data is non-uniformly partitioned through the Dirichlet distribution [34]. Finally, the data distributed to all participants involve uneven data samples, and the data sample sizes may vary widely among all participants. This is a more realistic scenario.

Model Architecture: We train the shared model as a classifier and classify the test set of the above two datasets. For different datasets, we have implemented two different Convolutional Neural Networks (CNN). We utilize two convolutional layers and one fully connected layer to train F-MNIST, while using six convolutional layers and two fully connected layers to

train CIFAR-10. Relu and Softmax are adopted as the activate functions of the convolutional layer and the output layer.

Training Settings: We use the following settings for the overall training process. First, we adopt the PyTorch (version 1.3.1) to build FL architecture in Ubuntu (version 18.04). Second, we set the number of participants in our joint training system to n = 50. Third, according to the rounds of training to the convergence state for the two datasets, we set the number of training rounds R for F-MNIST and CIFAR-10 to 100 and 200, respectively. Fourth, all participants (including malicious participants) train their data locally to build a local model, so that malicious participants can covertly perform label flipping poisoning attacks locally. Each malicious participant can virtual v sybil nodes to participate in the model aggregation. The system will randomly select o = 5 participants (including malicious adversaries) during aggregation to execute FEDAVG to build a shared model. Finally, for all the experiments, we perform distributed simulations on a single machine configuration with an Intel Xeon E5-2678 CPU, 32 GB RAM, and four NVIDIA GTX 1080 TI GPUs.

Attack Settings: In the baseline FL system without any malicious participants, we train a global model by using clean dataset and count the cases in which all classes are classified incorrectly during the testing under the conditions described in [35]. We record the global model accuracy and use this method as the baseline attack, and then compare the effect with our SCA. In order to eliminate the random influence of o in the aggregation stage, we repeated each experiment five times and calculated the average as the final result. Moreover, for the malicious adversary settings, we limit the total number of malicious participants that do not exceed half of the total participants ($k\%<50\%$), and each malicious participant can have no more than 10 virtual sybil nodes ($v<10$). For the label flipping attacks setting, we set two attack goals: Goal1: The easiest case to be misclassified in the test phase. We choose (6, 0) in the F-MNIST and (5, 3) for CIFAR-10. Goal2: The hardest case to be misclassified in the test phase. We choose (1, 3) in the F-MNIST and (0, 2) for CIFAR-10.

B. Performance Evaluation Metrics

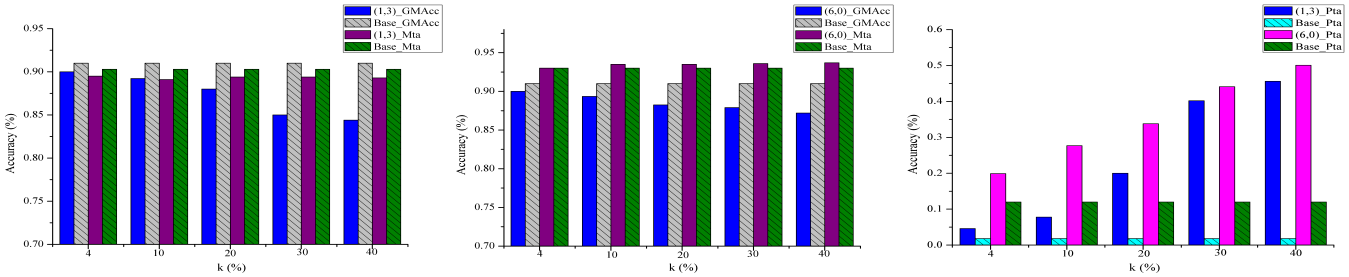
We utilize the two performance evaluation metrics to assess our proposed SCA. (1) Global Model Accuracy ($GMAcc$): we calculate the percentage of all correctly classified samples as the GMAcc value. (2) Attack Success Rate: this definition is used to assess the attack success rate of selected samples using the final poisoning global model obtained by our proposed SCA. Through the adversary objective mentioned above, we respectively define two performance metrics *poison task accuracy* (pta) and *main task accuracy* (mta) to show the attack effectiveness of our method more specifically. Where pta represents the percentage of source class samples classified as target class to the total samples of source class. Moreover, mta represents the percentage of samples from other non-attack classes that are correctly classified.

C. Performance of SCA

In evaluating the effectiveness of our SCA, we explore the impact of different k values and v values on attack performance, where k and v represent IIoT malicious devices and the virtualized malicious sybil nodes. Meanwhile, we verify the attack effectiveness of our SCA against non-iid data distribution. We finally analyze the convergence of the global model.

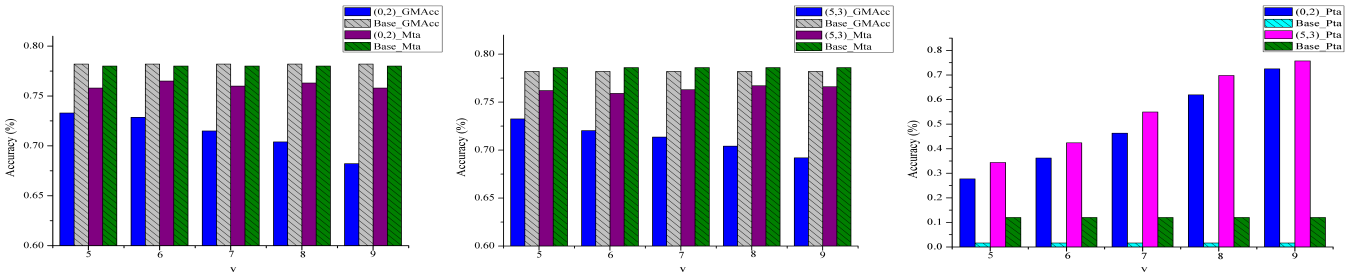
1) Evaluate the effectiveness of k: To demonstrate the performance impact of k on the attack effectiveness in our IIoT-FL system, we set k to a level lower than 50%, ranging from 4% to 40%. Moreover, each malicious participant only creates 5 sybil nodes. We consider that 4% is 2 malicious participants in our network. It is the case of the least number of malicious participants in a collusion attack. One can see from Fig.3, we can observe the performance metrics defined in the figure. $Base\_GMAcc$ represents the global model accuracy of the baseline attack in the convergence state, $Base\_Mta$ and $Base\_Pta$ are the main task accuracy and poison task accuracy calculated in this state. In Fig.3 (a) and Fig.3 (b), we can find that with the proportion of malicious participants k increases, the global model accuracy of $(1,3)\_GMAcc$ and $(6,0)\_GMAcc$ are gradually decreasing. Because the global model is gradually being guided by malicious models and is developing in a bad direction. Once k reaches the maximum value of 40%, this also produces the largest accuracy difference, and the result of performance degradation becomes more and more obvious. However, the main task accuracy calculated is different, because (6,0) is more likely to be misclassified than (1,3). The influence of the poisoning strategy makes the selected samples more likely to be misclassified, so $(6,0)\_Mta$ is higher than $(1,3)\_Mta$, but they all maintain similar accuracy as before, which shows that our attack will not interfere with the normal classification of other classes. Fig.3 (c) shows the poison task accuracy. It can be clearly observed that as k increases, $(1,3)\_Pta$ and $(6,0)\_Pta$ are gradually increasing. When k reaches the maximum value of 40%, the poisoning effect is most obvious. Even when k is 4%, our proposed attack strategy makes the global model produce a larger classification error rate than the baseline.

2) Evaluate the effectiveness of v: To verify the impact of the number of sybil nodes created by malicious participants on the model in our IIoT-FL system. We limit the number of v to less than 10, and we do not affect the global model performance by virtualizing too many sybils. In addition, the proportion of malicious participants k is set to 10% in the computing environment. We observe the attack effectiveness of our proposed SCA by creating different numbers of sybil nodes. One can see from Fig.4 (a) and Fig.4 (b), we observe the impact of v on the model performance by setting the range of v from 5 to 9. With the number of sybil nodes v gradually increasing, the global model accuracy of $(0,2)\_GMAcc$ and $(5,3)\_GMAcc$ gradually decreases. Because more and more sybils participate in model aggregation, and they inject poisoned local models into global models. When v reaches 9, the performance degradation becomes more significant. The main task accuracy remains similar to the baseline, with only 0.02% accuracy difference, which is acceptable. This shows that our SCA has little effect on other non-attack classes. Fig.4 (c) shows a more obvious effect in the poison task accuracy. Even if the system only sets 5 sybil nodes, the poison task accuracy



(a) GMAcc and Mta for F-MNIST(1,3). (b) GMAcc and Mta for F-MNIST(6,0). (c) Pta for F-MNIST(1,3) and F-MNIST(6,0).

Fig. 3. The performance results (GMAcc, Mta, and Pta) of F-MNIST(1,3) and F-MNIST(6,0) with different $k\%$. For each value of $k\%$, we set v = 5, and use the average of 5 runs as the final result.



(a) GMAcc and Mta for CIFAR-10(0,2). (b) GMAcc and Mta for CIFAR-10(5,3). (c) Pta for CIFAR-10(0,2) and CIFAR-10(5,3).

Fig. 4. The performance results (GMAcc, Mta, and Pta) of CIFAR-10(0,2) and CIFAR-10(5,3) with different v. For each value of v, we set $k\%$= 10%, and use the average of 5 runs as the final result.

of $(0,2)\_Pta$ and $(5,3)\_Pta$ is more than double the baseline. It shows that our proposed SCA makes the global model have a huge impact on the classification of specified attack class.

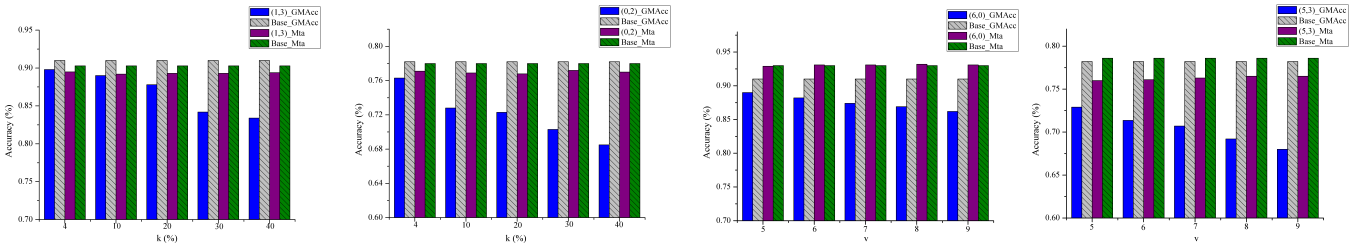
3) Effectiveness of non-iid data distribution: Devices in the IIoT-FL system are usually at risk of offline due to network (such as 5G, 4G, WIFI), communication delays, power supply, and other problems, which will result in non-iid distribution of data generated by many devices. To study the attack performance of our proposed SCA against non-iid data distribution, we utilize F-MNIST(1,3) and CIFAR-10(0,2) for testing (as k increases from 4% to 40%, v = 5), and utilize F-MNIST(6,0) and CIFAR-10(5,3) for testing (as v increases from 5 to 9, k = 10%). In our experimental setup, the malicious adversaries launch attacks according to the data distribution described in V-A. We show the $GMAcc$ and Mta of the two benchmarks in Fig.5. One can observe that as k increases, $GMAcc$ gradually decreases in (a) and (b), because the global model is gradually guided by the malicious model. When k reaches the maximum value of 40%, the difference in accuracy is more obvious. While Mta maintains similar performance as before, compared with the baseline, the maximum difference is within 0.05, which is acceptable. The same effect can also be found in (c) and (d). As v increases, the local poisoning model is aggregated with a greater probability, and the collusion model misclassifies more source class samples as target class. Meanwhile, more malicious adversaries participate in training. The poisoning parameters are rarely compromised by the parameters of clean data training during aggregation, which also potentially maintains the attack performance.

4) Convergence analysis: To verify the convergence of our proposed SCA, we use F-MNIST(6,0) and CIFAR-10(5,3) for testing in iid data distribution scenario, and use F-MNIST(1,3) and CIFAR-10(0,2) for testing in non-iid data distribution scenario. Specifically, for iid, the experimental setting is set to k = 10% and v = 5, and for non-iid, the experimental setting is set to k = 20% and v = 5, malicious participants launch the collusion attacks. We show the convergence curves of $GMAcc$ and Pta in all training rounds in Fig.6. It can be found that after the injection attack, the $GMAcc$ gradually increases with continuous training and is smaller than the baseline. Pta exhibits higher values with stronger attack effects, and finally converges. Our attack target is only for the selected source class, and does not interfere with other non-attack classes and the convergence of the global model. This result also shows that our SCA algorithm is effective in launching the poisoning collusion attack. Based on previous experimental studies and analysis, we propose the following proposition.

Proposition 1. Under the condition of limited malicious participants and the total number of virtual sybil nodes, the upper limit of the aggregated malicious model is $\frac{1}{o}\sum_{k=1}^{o} M_{Loc_{advk}}^{(r+1)}$.

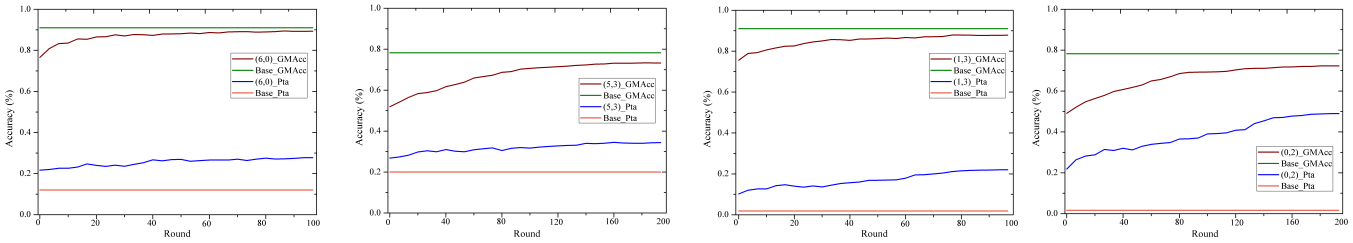
$$M_{Mal\_upp} = \frac{1}{o}\sum_{k=1}^{o} M_{Loc_{advk}}^{(r+1)} (k \rightarrow advk, advk \in o) \quad (7)$$

Due to the randomness of the participants o selected for aggregation in each update process, our SCA strategy aims to make more malicious adversaries selected to perform aggregation with greater probability. It is obvious that the global model is updated in the honest direction without aggregating malicious adversaries. When increasing the number of malicious participants or sybil nodes, the new global model will be guided by more malicious adversaries, who are updated in



(a) (GMAcc/Mta) for F-MNIST(1,3) (b) (GMAcc/Mta) for CIFAR-10(0,2) (c) (GMAcc/Mta) for F-MNIST(6,0) (d) (GMAcc/Mta) for CIFAR-10(5,3)

Fig. 5. The performance results (GMAcc and Pta) of F-MNIST(1,3) and CIFAR-10(0,2) with different k in (a) and (b) under non-iid data distribution. For each value of k, we set v = 5. The performance results (GMAcc and Pta) of F-MNIST(6,0) and CIFAR-10(5,3) with different v in (c) and (d) under non-iid data distribution. For each value of v, we set k= 10%, and use the average of 5 runs as the final result.



(a) iid F-MNIST(6,0) (b) iid CIFAR-10(5,3) (c) non-iid F-MNIST(1,3) (d) non-iid CIFAR-10(0,2)

Fig. 6. The convergence curves (GMAcc and Pta) for (a) iid F-MNIST(6,0), (b) iid CIFAR-10(5,3), (c) non-iid F-MNIST(1,3) and (d) non-iid CIFAR-10(0,2), and use the average of 5 runs as the final result.

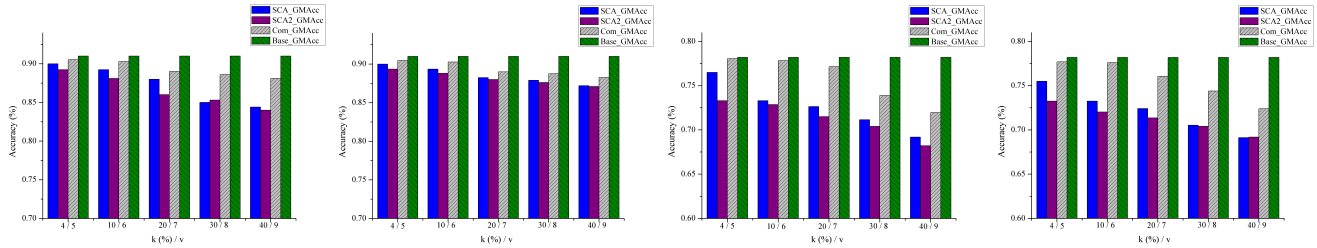
the malicious direction. When all the aggregated participants are malicious adversaries ($advk \in o$), the malicious model will compromise the update of the honest model, and finally aggregate into a purely malicious global model.

D. Comparison with the State-of-the-art

This module mainly compares our proposed SCA with the state-of-the-art in performance analysis (including $GMAcc$, $SCAcc$, aggregation possibility, and effectiveness of obscuring attack behavior). In the literature [16], the authors used label flipping attacks for local poisoning training in a federated environment, and committed to making the global model more direct to misclassify the attack targets. In order to compare with this method more intuitively and make a fairer performance analysis. We use the same set of participants, training dataset, and model architecture. Besides, we define a new evaluation metric, the *source class accuracy* ($SCAcc$), which represents the proportion of source class samples that can be correctly classified. We use it to verify the effectiveness of these attack methods against source class samples.

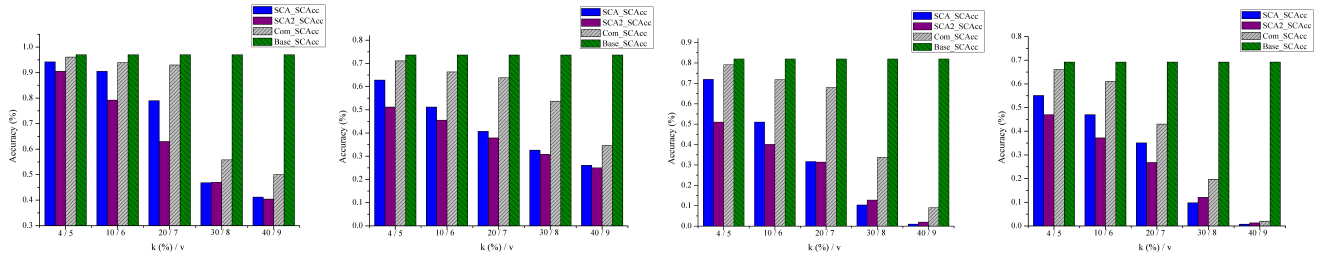
1) Evaluation of $GMAcc$: This section performs global model accuracy ($GMAcc$) analysis on the attack goals for two datasets. As shown in Fig.7, SCA_GMAcc represents the global model accuracy in the experimental setting (v = 5, k increased from 4% to 40%). $SCA2\_GMAcc$ represents the global model accuracy for (k=10%, v increased from 5 to 9). Com_GMAcc represents the global model accuracy of the comparison method for (k increased from 4% to 40%). $Base\_GMAcc$ represents the global model accuracy of the baseline attack. One can see from Fig.7 that compared to the baseline attack, the other three attack methods all show lower model accuracy. This is because affected by the attack of malicious participants, the server aggregates the poisoned models, which makes the global model misclassify the selected samples. Meanwhile, the accuracy of our proposed SCA under the two different experimental settings is lower than that of Com_GMAcc. The reason is that the sybils in the system help the local poisoning model to be aggregated by the global model with a greater probability. Although when the value of k/v is small, there is little difference in accuracy, with the increase of k/v, the difference in performance becomes more obvious. It can be found in Fig.7 (c) that when k = 20%, v = 5 the global model accuracy of SCA_GMAcc and the global model accuracy of $SCA2\_GMAcc$ obtained when k = 10%, v = 7 can be equivalent to the global model accuracy of Com_GMAcc under the setting of k = 40%. This result also shows that we only need a small number of malicious participants to achieve the attack effect obtained by a large number of malicious participants in the comparison method.

2) Evalution of $SCAcc$: We analyze the source class accuracy ($SCAcc$) of these attack methods in this section. Fig.8 summarizes the source class accuracy of SCA_SCAcc, $SCA2\_SCAcc$, Com_SCAcc, and $Base\_SCAcc$ respectively. It can be clearly found that compared to the baseline attack, the other three attack modes all show lower source class accuracy. Due to the influence of poisoning training on these selected attack classes, as the value of k/v increases, the global model will misclassify these samples more. Moreover, the source class accuracy of our proposed SCA under the two different experimental settings is lower than that of Com_SCAcc. The reason is that the sybil nodes in our computing environment make the local poisoning model more likely to be injected into the global model. This advantage is shown more clearly in Fig.8 (c) and Fig.8 (d), although



(a) MGAcc for F-MNIST(1,3) (b) MGAcc for F-MNIST(6,0) (c) MGAcc for CIFAR-10(0,2) (d) MGAcc for CIFAR-10(5,3)

Fig. 7. The performance results (MGAcc) of F-MNIST and CIFAR-10. We use the average of 5 runs as the final result.



(a) SCAcc for F-MNIST(1,3) (b) SCAcc for F-MNIST(6,0) (c) SCAcc for CIFAR-10(0,2) (d) SCAcc for CIFAR-10(5,3)

Fig. 8. The performance results (SCAcc) of F-MNIST and CIFAR-10. We use the average of 5 runs as the final result.

when k reaches 40%, SCA_SCAcc, $SCA2\_SCAcc$ and Com_SCAcc are basically reduced to 0. Because the global model is affected by collusion attacks launched by malicious participants in the system, and few source samples can be correctly classified in the testing phase.

TABLE I
PERFORMANCE RESULTS OF AGGREGATION POSSIBILITY.

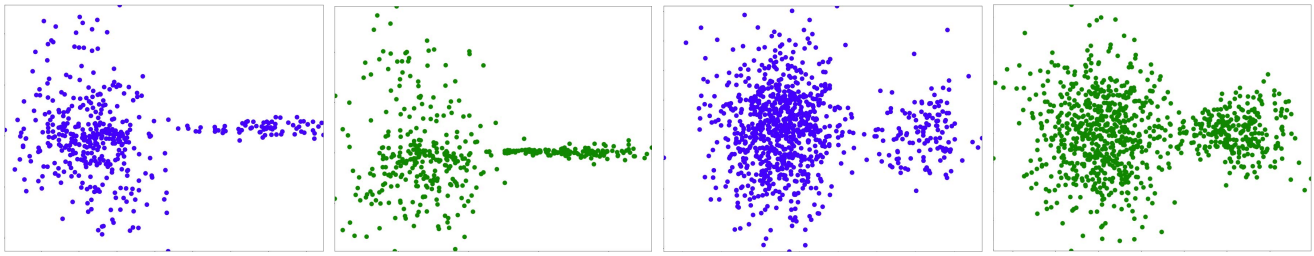
Dataset(F-MNIST&CIFAR-10)	SCA_ap	$SCA2\_ap$	Com_ap
k=4%,v=5 / k=10%,v=5	0.22	0.27	0.03
k=10%,v=5 / k=10%,v=6	0.27	0.34	0.11
k=20%,v=5 / k=10%,v=7	0.41	0.47	0.21
k=30%,v=5 / k=10%,v=8	0.58	0.56	0.3
k=40%,v=5 / k=10%,v=9	0.66	0.65	0.39

3) Evaluation of aggregation possibility: We select four experimental groups of the two datasets (F-MNIST(1,3), F-MNIST(6,0), CIFAR-10(0,2), and CIFAR-10(5,3)), and count the total malicious participants chosen during aggregation. The R of F-MNIST and CIFAR-10 are 100 and 200, respectively. Five participants (including malicious adversaries) are selected for aggregation in each round. Finally, we calculate the aggregation probability of malicious adversaries in each set of experiments, denoted as the proportion of the total number of malicious adversaries counted in the total aggregation of participants. SCA_ap, $SCA2\_ap$, Com_ap and Com_ap in Table I are represented the aggregation possibility in above experimental settings. Each result in Table I is the average value calculated for four experimental groups under the same experimental setting. We can find that compared with Com_ap, our proposed SCA has a higher aggregation possibility, because sybils play an essential role in the aggregation process. Even if we use a small part of malicious participants to join the training, we can achieve similar effects when the comparison method uses many malicious participants.

4) Effectiveness of obscuring attack behavior: Malicious adversaries utilize the vulnerability of FL aggregation to upload and aggregate the poisoned local model, aiming to damage the global model's performance. Generally, anomaly detection methods (such as identifying malicious updates [30]) are used to detect the uploaded anomalous parameters. To show the characteristics of our collusion attack method more reliably, we respectively use gradient detection to verify the parameter update of the two attack methods. We aim to demonstrate that our collusion attacks can effectively obscure their attack behavior against anomaly detection. In addition, for a fair comparison, we select two sets of experiments (F-MNIST(6,0) and CIFAR-10(0,2) with the experimental setting k = 10%, v = 6 for SCA, and k = 30% for Com) with similar aggregation possibility shown in Table I for verification. One can see from Fig.9, we visualize the gradient value of our SCA in blue, and the other in green. It can be found from Fig.9 (a) that the abnormal gradient is displayed on the right side of the picture, which is the result of our counting. The abnormal gradient detected is less than the count in Fig.9 (b). Similar performance comparison results are also can be found in Fig.9 (c) and Fig.9 (d). Since the CIFAR-10 dataset runs for 200 rounds, the effect is more obvious. Compared with the independent attacks performed in Fig.9 (b) and Fig.9 (d), the collusion attacks we implemented perform a gradient fine-tuning of the uploaded poisoning model. Our proposed SCA can effectively obscure their attack behavior, and further destroy the performance of the global model after aggregation.

VI. CONCLUSION

This paper analyzed the security vulnerabilities of joint training in the IIoT-FL system, then proposed a sybil-based collusion attacks (SCA) approach for the vulnerabilities. Meanwhile, we also gave further details on the execution of



(a) F-MNIST(6,0) for SCA (b) F-MNIST(6,0) for Com (c) CIFAR-10(0,2) for SCA (d) CIFAR-10(0,2) for Com

Fig. 9. An overview of performance results for F-MNIST(6,0) and CIFAR-10(0,2). Assess the effectiveness of obscuring their attack behavior for two attack methods. The gradients of SCA we highlight in blue, and the other highlight in green.

related algorithms, model architecture, and analysis of the effectiveness of the experiment. In this work, malicious participants in our federated system can virtualize multiple sybil nodes and perform malicious collusion attacks. The purpose is to make the local poisoning model be aggregated with a greater possibility. They aim to make the samples of the selected attack class be misclassified, while other non-attack classes maintain similar accuracy as before. Compared with the state-of-the-art, our SCA can achieve a more substantial attack effect under the condition of fewer malicious participants performing collusion, and can successfully obscure their attack behavior. Extensive experimental results show that our SCA has a more robust attack performance on several evaluation metrics.

REFERENCES

[1] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, D. Niyato and H. V. Poor, "Federated learning for industrial internet of things in future industries," *IEEE Wireless communications magazine*, 2021.

[2] 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 (TII)*, 2021.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273-1282.

[4] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: A big data - AI integration perspective," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 33, no. 4, pp. 1328-1347, 2021.

[5] B. Jia, X. Zhang, J. Liu, Y. Zhang, K. Huang, and Y. Liang, "Blockchain-enabled federated learning data protection aggregation scheme with differential privacy and homomorphic encryption in IIoT," *IEEE Transactions on Industrial Informatics (TII)*, 2021.

[6] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *ELSEVIER Future Generation Computer Systems (FGCS)*, vol. 115, pp. 619-640, 2021.

[7] T. Li, A. K. Sahu, A. Talwalker, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50-60, 2020.

[8] W. S. Zhang, T. Zhou, Q. H. Lu, X. Wang, C. S. Zhu, H. Y. Sun, Z. P. Wang, S. K. Lo, and F. Y. Wang, "Dynamic fusion-based federated learning for COVID-19 detection," *IEEE Internet of Things Journal (IoTJ)*, 2021.

[9] M. Parimala, M. S. Swarna, P. V. Quoc, D. Kapal, M. Praveen, T. Gadekallu, and T. H. Thien, "Fusion of federated learning and industrial internet of things: A survey," arXiv preprint arXiv:2101.00798, 2021.

[10] M. X. Duan, K. L. Li, A. J. Ouyang, K. N. Win, K. Q. Li and Q. Tian, "EGroupNet: A feature-enhanced network for age estimation with novel age group schemes," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 16, no. 2, 2020.

[11] U. M. Aivodji, S. Gambs, and A. Martin, "IOTFLA : A secured and privacy-preserving smart home architecture implementing federated learning," in *Proceedings of the 2019 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2019, pp. 175-180.

[12] Y. Song, T. Liu, T. Wei, X. Wang, Z. Tao, and M. Chen, "FDA3 : Federated defense against adversarial attacks for cloud-based IIoT applications," *IEEE Transactions on Industrial Informatics (TII)*, 2020.

[13] M. Rehman,and A. Dirir. "TrustFed: A framework for fair and trustworthy cross-device federated learning in IIoT," *IEEE Transactions on Industrial Informatics (TII)*, 2021.

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

[15] E. Bagdasaryan, A. Veit, Y. Q. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS)*, PMLR, 2020, pp. 2938-2948.

[16] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, Springer, 2020, pp. 480-501.

[17] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2019.

[18] Y. Jiang, S. Q. Wang, V. Valls, B. J. Ko, W. H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," arXiv preprint arXiv:1909.12326, 2019.

[19] Y. Qu, S. Pokhrel, S. Gary, L. Gao, and Y. Xiang. "A blockchained federated learning framework for cognitive computing in industry 4.0 networks," *IEEE Transactions on Industrial Informatics (TII)*, vol. 17, no. 4, pp. 2964-2973, 2020.

[20] M. H. Fang, X. Y. Cao, J. Y. Jia, and N. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *Proceedings of the 29th USENIX Security Symposium (USENIX Security)*, USENIX, 2020.

[21] Y. Liu, R. H. Zhao, J. W. Kang, A. Yassine, D. Niyato, and J. L. Peng, "Towards communication-efficient and attack-resistant federated edge learning for industrial internet of things," arXiv preprint arXiv:2012.04436, 2020.

[22] C. L. Xie, K. L. Huang, P. Y. Chen, and B. Li, "DBA: Distributed backdoor attacks against federated learning," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

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

[24] C. Fung, C. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," arXiv preprint arXiv:1808.04866, 2018.

[25] Y. P. Jiang, Y. Li, Y. P. Zhou, and X. Zheng, "Mitigating sybil attacks on differential privacy based federated learning," arXiv preprint arXiv:2010.10572, 2020.

[26] C. Fung, C. M. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *Proceedings of the 23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID)*, USENIX, 2020.

[27] A. K. Mishra, A. K. Tripathy, D. Puthal, and L. T. Yang, "Analytical model for sybil attack phases in internet of things," *IEEE Internet of Things Journal (IoTJ)*, vol. 6, no. 1, pp. 379-387, 2019.

[28] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "FED-IIoT: A robust federated malware detection architecture in industrial IoT," *IEEE Transactions on Industrial Informatics (TII)*, 2020.

[29] W. B. Lim, N. C. Luong, D. T. Hoang, Y. T. Jiao, and Y. C. Liang, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031-2063, 2020.
[30] N. Baracaldo, B. Chen, H. Ludwing, and J. A. Safavi, "Mitigating poisoning attacks on machine learning models: A data provenance based approach," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec)*, ACM, 2017, pp. 103-110.
[31] Z. Y. Du, C. Wu, T. Yoshinaga, Y. S. Ji, and J. Li, "Federated learning for vehicular internet of things: Recent advances and open issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45-61, 2020.
[32] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," arXiv preprint arXiv:1708.07747. https://github.com/zalandoresearch/fashion-mnist
[33] A. Krizhevsky, and G. Hinton, "Learning multiple layers of features from tiny images." http://www.cs.toronto.edu/∼ kriz/cifar.html
[34] T. Minka. "Estimating a Dirichlet distribution." Technical report, MIT, 2000.
[35] D. Cao, S. Chang, Z. J. Lin, G. H. Liu, and D. H. Sun, "Understanding distributed poisoning attack in federated learning," in *Proceedings of the 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2019.

Xiong Xiao is currently working toward the PhD degree at the College of Computer Science and Electronic Engineering, Hunan University, China. His main research interests include cloud computing, scheduling for distributed computing systems, distributed machine learning and privacy and security of federated learning.

Zhuo Tang received the Ph.D. in computer science from Huazhong University of Science and Technology, China, in 2008. He is currently a professor of the College of Computer Science and Electronic Engineering at Hunan University. He is also the chief engineer of the National Supercomputing Center in Changsha. His majors are distributed computing system, cloud computing, and parallel processing for big data, including distributed machine learning, security model, parallel algorithms, and resources scheduling and management in these areas. He has published almost 90 journal articles and book chapters. He is a member of IEEE/ACM and CCF.

Chuanying Li is currently working toward the PhD degree at the College of Computer Science and Electronic Engineering, Hunan University, China. Her research interests include high performance computing and cloud computing.

Bin Xiao received the B.Sc. and M.Sc. degrees in electronics engineering from Fudan University, and the Ph.D. degree in computer science from The University of Texas at Dallas, USA. He is currently an Associate Professor with the Department of Computing, The Hong Kong Polytechnic University. Dr. Xiao has over fifteen years research experience in the cyber security, and currently focuses on the network and cloud security, blockchain technology and AI security. He published more than 150 technical papers in international top journals and conferences. Currently, he is the associate editor of IEEE TCC, IEEE TNSE, IEEE IoTJ, and Elsevier JPDC. He is the vice chair of IEEE ComSoc CISTC committee. He has been the symposium co-chair of IEEE ICC2020, ICC 2018 and Globecom 2017, and the general chair of IEEE SECON 2018.

Kenli Li received the PhD degree in computer science from Huazhong University of Science and Technology, in 2003. He was a visiting scholar at University of Illinois at Urbana-Champaign from 2004 to 2005. He is currently a full professor of computer science and technology at Hunan University and deputy director of National Supercomputing Center in Changsha. His major research includes parallel computing, cloud computing, and Big Data computing. He has published more than 300 papers in international conferences and journals. He serves on the editorial boards of IEEE Transactions on Computers, IEEE Transactions on Industrial Informatics, IEEE Transactions on Sustainable Computing, International Journal of Pattern Recognition and Artificial Intelligence. He is a senior member of IEEE and an outstanding member of CCF.