Sybil-based Virtual Data Poisoning Attacks in Federated Learning*

Changxun Zhu[1], Qilong Wu[1], Lingjuan Lyu[2] and Shibei Xue[1]

Abstract— **Federated learning is vulnerable to poisoning attacks by malicious adversaries. Existing methods often involve high costs to achieve effective attacks. To address this challenge, we propose a sybil-based virtual data poisoning attack, where a malicious client generates sybil nodes to amplify the poisoning model's impact. To reduce neural network computational complexity, we develop a virtual data generation method based on gradient matching. We also design three schemes for target model acquisition, applicable to online local, online global, and offline scenarios. In simulation, our method outperforms other attack algorithms since our method can obtain a global target model under non-independent uniformly distributed data.**

Keywords—Federated learning, Sybil poisoning attack, Virtual data.

I. INTRODUCTION

The revolution in sensing technology has enabled high-quality data acquisition and processing across diverse real-world applications. This technological progress has catalyzed significant advancements in artificial intelligence (AI), achieving state-of-the-art performance in specialized domains including natural language processing [1], recommender systems [2], [3], pose estimation [4], [5], intelligent transportation [6], [7], energy-related prediction [8]–[10].

However, with the growing emphasis on data privacy and the introduction of data protection regulations, traditional centralized machine learning approaches face significant obstacles [11]. To address this, federated learning (FL) [12] emerges as a privacy-preserving paradigm. FL establishes a shared model on a central server, distributes the model to clients for training on local data, and subsequently aggregates the locally trained models on the server. This framework avoids direct data transmission, thereby preserving client privacy [13].

While federated learning preserves data locality on client devices, it also introduces new challenges. The inability to filter user data results in non-independent and identically distributed (Non-IID) data, leading to model drift and prolonged convergence times for optimal performance [14]. Additionally, the lack of data filtering makes federated learning susceptible to attacks by malicious adversaries.

To address the aforementioned challenges, designing federated learning defense algorithms to enhance the stability of the federated learning process is a practical approach [15]. Another perspective is to deepen the study of federated learning attack algorithms to understand potential security risks, thereby improving the security and privacy protection of federated learning. The latter can better understand the attack process from the attacker's perspective, which is more conducive to our formulation of proactive defense strategies.

The earliest poisoning attack was introduced against Support Vector Machines (SVM) by flipping the labels of training data [16]. Although originally designed for centralized settings, this attack is found to be effective in federated learning scenarios [17]. Based on the structural characteristics of federated learning, the following poisoning attacks can be categorized into three types: data poisoning, model poisoning, and sybil-based poisoning attacks.

In data poisoning attacks, adversaries cannot directly manipulate users' models but can access and tamper with client training data to execute attacks. Ref. [17] first introduced label-flipping attacks to federated learning, where malicious actors flip sample labels, causing the trained model to deviate from the intended prediction boundary. However, the effectiveness of this approach is limited by the influence of non-malicious clients. To address this, Ref. [18] proposed a dynamic label-flipping strategy that selects the target label with the smallest loss, improving on static label-flipping methods. Beyond label-flipping attacks, clean-label poisoning is another common data poisoning approach. This technique retains original labels but injects malicious patterns into model parameters through image pixel optimization [16]. However, this approach is computationally expensive for deep neural networks. To overcome this limitation, heuristic methods have been proposed, as demonstrated in Refs. [19], [20], to achieve clean-label poisoning more efficiently.

However, the success rate of data poisoning attacks is directly proportional to the number of malicious clients controlled by the attacker, making such attacks costly in large-scale federated learning systems. To address this limitation, model poisoning attacks were introduced, enabling adversaries to manipulate the local training process. Ref. [21] demonstrated an attack executed when the global model nears convergence, modifying the local training process by adding an anomaly detection term to the loss function. In contrast, Ref. [22] proposed an attack targeting the early stages of global model training, before convergence is achieved. Additionally, Ref. [23] leveraged a regularization term in the objective function to embed malicious neurons into the redundant spaces of neural networks. This approach minimizes the impact of benign clients during model aggregation, allowing the attacker to execute poisoning attacks effectively.

Sybil-based attacks are another common method of dis-

*This work was supported in part by the National Natural Science Foundation of China under Grant 62273226 and Grant 61873162. (Corresponding author: Shibei Xue)

[1]Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, P.R. China (e-mail: shbxue@sjtu.edu.cn).

[2]Sony AI

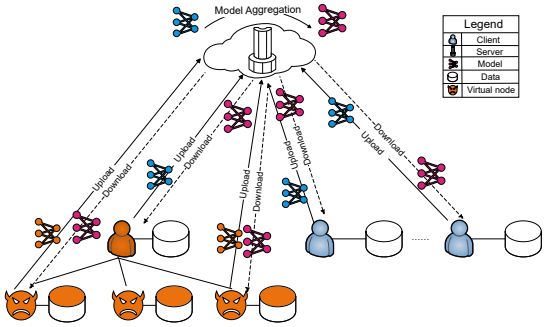


Fig. 1 Threat model for sybil-based attacks on FL.

rupting federated learning. In such attacks, a malicious attacker creates multiple fake clients to manipulate the model's learning process, potentially causing training errors or compromising private information. Ref. [24] first introduced sybil-based attacks in federated learning, proposing a novel denial-of-service (DoS) attack. Inspired by this work, Refs. [25], [26] employed sybil node collusion strategies to enhance attacker cooperation, enabling more effective poisoning attacks such as label flipping and backdoor attacks. However, the existing sybil-based poisoning attack methods often rely on sharing client data, which is costly and impractical in privacy-sensitive federated learning environments.

To address this challenge, we propose a sybil-based virtual data poisoning attack. Instead of sharing data with sybil nodes directly, we heuristically propose a novel approach for generating training data for sybil nodes based on gradient matching, which effectively reduces both the complexity of solving the optimization problem and the computational cost. Furthermore, we introduce three adaptive model acquisition strategies tailored to distinct deployment scenarios, enabling precise manipulation of model update directions. Experimental results demonstrate that our method significantly enhances the effectiveness of poisoning attacks compared to baseline methods.

The structure of this paper is as follows: Section II provides a brief overview of problem description and the sybil-based attack model in FL, followed by a detailed explanation of our proposed method in Section III. Section IV presents the experimental setup and results. Finally, Section V concludes the paper.

II. PROBLEM DESCRIPTION OF SYBIL-BASED ATTACK

In federated learning, data owners are referred to as clients, denoted by $\mathscr{C} = \{C_1, C_2, \ldots, C_N\}$, where N represents the total number of clients. Each client C_i has access to its local training dataset D_i. A central server oversees model initialization, aggregation, and distribution. The server begins by initializing a model w_0 and distributing it to all clients. Each client C_i trains the model locally using its dataset D_i and sends the updated model back to the server for aggregation.

The sybil-based attack model is illustrated in Fig. 1. In this model, attackers control a subset of malicious clients with real data to generate poison data. In a federated learning system with N clients, we assume that the attacker can control $m\%$ of the clients as malicious and generate v sybil nodes for each malicious client. The total number of sybil nodes, denoted as $|\mathscr{V}|$, is given by:

$$|\mathscr{V}| = N * m\% * v. \tag{1}$$

To attack an image classification task, the objective is to train a model using poisoned data to misclassify images from a target category, while preserving the accuracy of classification for other categories of images. Assume there are N_1 poisoning training samples from a target category and N_2 samples with normal labels. Taking the label-flipping attack as an example, we flip the labels of the target category from y_i^{tar} to y_i^{adv}. Thus, our optimization problem can be formulated as follows:

$$\min_{\Delta \in \mathscr{C}} \mathbf{J} = \sum_{i=1}^{N_1} l\left(f_{w(\Delta)}(x_i), y_i^{adv}\right) + \sum_{i=1}^{N_2} l\left(f_{w(\Delta)}(x_i), y_i\right), \tag{2}$$

$$s.t.\ w(\Delta) \in \arg\min_{w} \bar{\mathbf{J}} = \frac{1}{P} \sum_{i=1}^{P} l\left(f_w(x_i + \Delta_i), y_i\right), \tag{3}$$

$$C = \left\{\Delta \in R^{P \times n} : \|\Delta\|_\infty \leq \varepsilon\right\}, \tag{4}$$

The first term in Eq. (2) captures the error resulting from the misclassification of the target class, while the second term accounts for the error in correctly classifying images from other categories. $f_{w(\Delta)}(x_i)$ represents the probability that the model $w(\Delta)$ assigns a category prediction to x_i. Eq. (3) defines the constraints for training with poisoned data on sybil nodes, where P denotes the number of poisoned images and Δ represents the perturbation introduced during the poisoning process. Eq. (4) constrains the perturbations, with C denoting the set of perturbations and ε indicating the perturbation threshold. l representing the cross-entropy loss function, is written as

$$l(p,q) = \sum_{i=1}^{n} p(x_i) \log \frac{1}{q(x_i)} = -\sum_{i=1}^{n} p(x_i) \log q(x_i). \tag{5}$$

It is evident that our optimization problem forms a bilevel optimization structure. We refer to the objective function of the inner optimization, shown in Eq. (3), as the training loss, and the objective function of the outer optimization, shown in Eq. (2), as the adversarial loss.

III. METHOD

This section is divided into four subsections to detail the core components of the attack strategy. First, we introduce three approaches for obtaining the target model and a method for acquiring the baseline dataset. Next, to address the complexity of the task, we simplify the bilevel optimization problem using a gradient matching technique. After that, we describe the process of generating virtual poisoning data for sybil nodes to carry out poisoning attacks. Finally, we briefly introduce the overall algorithm workflow.

A. Acquisition of Target Model and Baseline Dataset

In each round of the federated learning process, it is essential to determine the update direction of the poisoning model by obtaining the target model in advance. Additionally, we explain how to acquire a benchmark dataset for training malicious clients in this subsection.

To identify the target image with the label y^{adv}, we select images labeled as y^{adv} from the controlled client's dataset as the baseline dataset D_i^{base}. Specifically, D_i^{base} is defined as shown in Eq. (6), where D_i represents the local dataset of the controlled clients.

$$D_i^{base} = \left\{ (x,y) \mid (x,y) \in D_i, y = y^{adv} \right\}. \tag{6}$$

To obtain the target model, we propose specific schemes for three scenarios: local online, global online, and offline. In the online local target model acquisition scheme, the attacker performs a fake local training using the global model w distributed by the server. The training data is defined in Eq. (7). Using

$$D_i^{mdf} = \left\{ (x_i, y_i') \mid (x_i, y_i) \in D_i, y_i' = \begin{cases} y^{adv}, & \text{if } y_i = y^{tar} \\ y_i, & \text{otherwise} \end{cases} \right\}, \tag{7}$$

the attacker trains a poisoning model w_i^{mdf}, which serves as the target model w^{tar}. Thus, $w^{tar} = w_i^{mdf}$.

Due to the non-IID nature of data across clients, obtaining the target model from a single client may hinder effective poisoning attacks. Some malicious clients may lack data for the target class, making label flipping infeasible. To address this, the online global target model scheme aggregates the w_i^{mdf} models generated by all controlled malicious clients. The target model is defined as

$$w^{tar} = w^{mdf} = \sum_{i=1}^{M} \frac{1}{M} w_i^{mdf}. \tag{8}$$

Obtaining the target model during the poisoning process of federated learning demands excessive real-time communication resources. To address this, we propose an offline target model acquisition scheme which is implemented before federated learning begins. The attacker distributes an untrained model to each controlled malicious client and uses their local label-flipped dataset D_i^{mdf} for R rounds of training and aggregation. The final aggregated model is used as the target model w^{tar}.

B. Problem Simplification by Gradient Matching

Given the complexity of our task, solving the optimization problem solely with neural networks is challenging. To address this, we simplify the calculation process using a gradient matching method.

In our bilevel optimization problem, the goal is to ensure that both the training and adversarial losses decrease concurrently through gradient descent. This allows the two objective functions to reach their low-value regions simultaneously. We thus have

$$\frac{1}{N_1+N_2}\left(\sum_{i=1}^{N_1} \nabla l\left(f_w(x_i), y_i^{adv}\right) + \sum_{i=1}^{N_1} \nabla l\left(f_w(x_i), y_i\right)\right) \approx \frac{1}{P}\sum_{i=1}^{P} \nabla_w l\left(f_w(x_i+\Delta_i), y_i\right). \tag{9}$$

However, finding poisoning images that satisfy Eq. (9) is challenging throughout the gradient descent process. In this case, we relax the requirement for gradient matching and instead aim to make the gradients of the model with respect to the objective functions of the inner and outer optimizations as similar as possible. Therefore, the bilevel optimization problem in Eqs. (2), (3), (4) can be heuristically rewritten as

$$\nabla \mathbf{J} = \textstyle\sum_{i=1}^{N_1} \nabla l\left(f_w(x_i), y_i^{adv}\right) + \sum_{i=1}^{N_2} \nabla l\left(f_w(x_i), y_i\right), \tag{10}$$

$$\nabla \bar{\mathbf{J}} = \textstyle\sum_{i=1}^{P} \nabla_w l\left(f_w(x_i+\Delta_i), y_i\right), \tag{11}$$

$$\mathscr{B}(\Delta; w) = 1 - \frac{\left\langle \nabla \mathbf{J}, \nabla \bar{\mathbf{J}} \right\rangle}{\|\nabla \mathbf{J}\| \cdot \left\|\nabla \bar{\mathbf{J}}\right\|}, \tag{12}$$

respectively.

We reformulate the original optimization problem using the form of negative cosine similarity, as shown in Eq. (12), where $\|\cdot\|$ and $\langle\cdot\rangle$ represent the L_2 norm and dot product of vectors. By minimizing Eq. (12), the value of the second term will gradually approach 1. According to the cosine similarity relationship, $\nabla \mathbf{J}$ and $\nabla \bar{\mathbf{J}}$ will become as similar as possible.

C. Generation of Poisoning Data by Sybil Nodes

To address the potential misalignment of gradient descent directions caused by mini-batch training in federated learning, we approximate the gradient using the negation of the difference between the target model w^{tar} and the global model at the current round w^r. This approach provides a more representative descent direction as

$$\sum_{i=1}^{N_1} \nabla l(f_{w^r}(x_i), y_i^{adv}) + \sum_{i=1}^{N_2} \nabla l(f_{w^r}(x_i), y_i) \approx w^r - w^{tar}. \tag{13}$$

Consequently, Eq. (12) is reformulated as

$$\mathscr{B}(\Delta; w^r) = 1 - \frac{\left\langle w^r - w^{tar}, \sum_{i=1}^{P} \nabla_{w^r} l(f_{w^r}(x_i+\Delta_i), y_i) \right\rangle}{\|w^r - w^{tar}\| \cdot \left\|\sum_{i=1}^{P} \nabla_{w^r} l(f_{w^r}(x_i+\Delta_i), y_i)\right\|}, \tag{14}$$

where (x_i, y_i) are sampled from D_i^{base}.

As the local training process is immutable, the attacker can only introduce poisoned data through virtual nodes controlled by malicious clients. Using the target model w^{tar} and the baseline dataset D^{base}, Eq. (14) is optimized via stochastic gradient descent, with the perturbation vector Δ updated as

$$\Delta(t+1) = \Delta(t) - \nabla_{\Delta(t)} \mathscr{B}. \tag{15}$$

The resulting poisoning data

$$D_{poison}^{base} = \left\{ (x', y) \mid (x, y) \in D_i^{base}, x' = x + \Delta_i \right\} \tag{16}$$

is then injected into the sybil nodes.

D. Complete Process

In summary, our sybil-based poisoning attack algorithm is described in Algorithm 1. In each training round, a subset S_r is selected from the entire set of clients, including both real clients and virtual sybil nodes. Different training strategies are applied based on the client type within S_r. Benign clients perform standard local training, whereas malicious clients first obtain the target model and then generate poisoning data using their local dataset for training sybil nodes. The sybil nodes utilize the poisoning data generated by malicious clients for training and subsequently upload the poisoning models to the central server, where they contribute to the aggregation and update of the global model.

IV. EXPERIMENT

A. Experimental Setup

Datasets. To comprehensively evaluate the performance of our proposed algorithm on image classification tasks, we conduct experiments on three widely used datasets in federated learning: MNIST [27] (70K handwritten digit images, 10 classes), FMNIST [28] (70K fashion item images, 10 classes), and CIFAR-10 [29] (70K images, 10 classes). For MNIST and FMNIST, we use the online global target model acquisition scheme, while for CIFAR-10, we adopt the offline target model acquisition scheme.

Networks. We design training networks of varying complexity tailored to each dataset. For MNIST, a fully connected neural network (FC) with four layers is employed, the details of which are shown in Table I. The final layer employs cross-entropy loss without activation for multi-classification. For FMNIST, a convolutional neural network (CNN) is employed. Details are provided in Table II. A ReLU activation function [30] follows each hidden layer. For CIFAR-10, the ResNet18 architecture is utilized.

TABLE I: Fully connected neural network structure table

Layers	Layer type	Input size	Output size
1	FC	[784]	[32]
2	FC	[32]	[16]
3	FC	[16]	[8]
4	FC	[8]	[10]

TABLE II: Convolutional neural network structure table

Layers	Layer type	Input size	Output size
1	CONV	[1,28,28]	[6,24,24]
2	MaxPool	[6,24,24]	[6,12,12]
3	CONV	[6.12,12]	16,8,8]
4	MaxPool	[16,8,8]	[16,4,4]
5	FC	[16,4,4]	[120]
6	FC	[120]	[84]
7	FC	[84]	[10]

Environment setting. We implement the federated learning process using PyTorch (version 1.13.0) in a distributed training setup. For local training on clients, we employ the SGD optimizer with a learning rate of $\eta = 0.01$ and momentum of 0.9. Each client train for $E = 5$ cycles per round with a batch size of $B = 64$. The total number of clients

Algorithm 1 Sybil-based poisoning attack algorithm

Input: Communication rounds, R; The number of clients, N; Local training rounds, E; Learning rate, η;
Output: Final model, w^R;

Server execution:
Initialize w^0
for $r \leftarrow 0$ **to** $R-1$ **do**
  Select the client set S_r from the $N + v*M$ clients
  for $i \leftarrow 1$ to $|S_r|$ parallel execution **do**
    Send the global model w^r to the client C_i
    if $M \le i \le N$ **then**
      $w_i^r \leftarrow$ Client local training (i, w_r)
    else if $1 \le i \le M$ **then**
      $w^{tar} \leftarrow$ Target model acquisition(w^r)
      $D_{poi}^{base} \leftarrow$Poisoning data generation(D_i^{base}, w^{tar})
    else if $1+N \le i \le N+v*M$ **then**
      Obtain D_i from the malicious client
      $w_i^r \leftarrow$ Client local training (i, w_r)
    end if
  end for
  Server update: $w^{r+1} = \sum_{i \in S_r} \frac{|D_i|}{D_{S_r}} w_i^r$
end for
return w^R
Client local training (i, w_r)
$w_i^r(0) \leftarrow w^r$
for $t \leftarrow 0$ **to** $\tau = \frac{|D_i|}{B} E - 1$ **do**
  $l_i(w_i^r(t)) \leftarrow \text{CrossEntropyLoss}\left(f_{w_i^r(t)}(x), y\right)$
  $w_i^r(t+1) \leftarrow w_i^r(t) - \eta \nabla l_i(w_i^r(t))$
end for
$w_i^r \leftarrow w_i^r(\tau)$
Send w_i^r back to the server.
Poisoning data generation
$\Delta(0) \leftarrow 0$
for $t \leftarrow 0$ **to** $T-1$ **do**
  $\mathscr{B}(\Delta; w^r) \leftarrow 1 - \frac{\left\langle w^{tar} - w^r, \sum_{i=1}^{P} \nabla_{w^r} l(f_{w^r}(x_i + \Delta_i), y_i) \right\rangle}{\|w^{tar} - w^r\| \cdot \left\| \sum_{i=1}^{P} \nabla_{w^r} l(f_{w^r}(x_i + \Delta_i), y_i) \right\|}$
  $\Delta(t+1) \leftarrow \Delta(t) - \nabla_{\Delta(t)} \mathscr{B}$
end for
$D_{poison}^{base} \leftarrow \left\{ (x', y) \mid (x, y) \in D_i^{base}, x' = x + \Delta_i \right\}$
Send D_{poison}^{base} to clients $C_{N+i}, \cdots, C_{N+v}$

$N = 50$, all of which participate in every communication round. The number of communication rounds is set to $R = 300$ for MNIST and FMNIST, and $R = 200$ for CIFAR-10, at which point the federated averaging algorithm achieve stable accuracy. Based on experience, attacks are executed during the final 50 rounds for all three datasets. Simulations were conducted using an Intel Xeon Platinum 8255C CPU, 40 GB RAM, and four NVIDIA RTX 3080 GPUs on a single machine configuration.

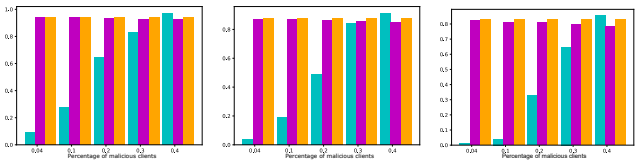
Parameters. By default, the number of malicious clients is $M = 20$ (40% of the total). Each client generates $v = 5$ sybil nodes, and the Dirichlet distribution hyperparameter is

set to $\alpha = 0.5$. The disturbance vector Δ is unconstrained in size, and poisoning data generation halts after a fixed number of iterations, set to $T = 300$, with a learning rate of 1. For each malicious client, we compute the disturbance vector for only 32 images from its baseline data. By default, the goal is to misclassify data labeled as '1' into '7', making the target category 1, the adversarial category 7, and the baseline category also 7.

Evaluation Metrics. This study employs Main Task Accuracy (MTA) and Target Task Accuracy (TTA) as evaluation metrics. In our study, TTA measures the accuracy of the global model in classifying target images into the desired category. MTA evaluates the accuracy of the global model in classifying non-target images into their original categories. Both of these metrics are evaluated under poisoning attacks. Additionally, we compare MTA with the global model accuracy (GMA) which is computed in the absence of malicious clients.

B. Effect of the Number of Sybil Nodes

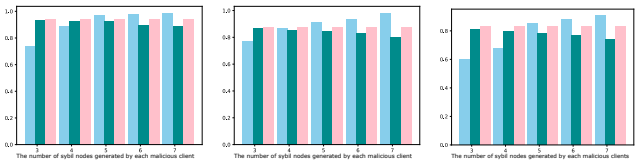
This experiment investigates the impact of the proportion of malicious clients and the number of sybil nodes generated by each on the poisoning attack. First, we fix the proportion of malicious clients at $m\% = 40\%$ and vary the number of sybil nodes generated by each malicious client $v = \{5,6,7,8,9\}$. After that, we fix $v = 5$ and vary the proportion of malicious clients $m\% = \{4\%,10\%,20\%,30\%,40\}$. In both cases, attacks are launched in the last 50 rounds across all three datasets. The results are shown in Fig.2 and Fig.3.



(a) MNIST dataset **(b)** FMNIST dataset **(c)** CIFAR-10 dataset

TTA of our method MTA of our method GMA(without malicious clients)

Fig. 2 The test accuracy changes with the proportion of malicious clients while fix the number of virtual nodes generated by each client v = 5.



(a) MNIST dataset **(b)** FMNIST dataset **(c)** CIFAR-10 dataset

TTA of our method MTA of our method GMA(without malicious clients)

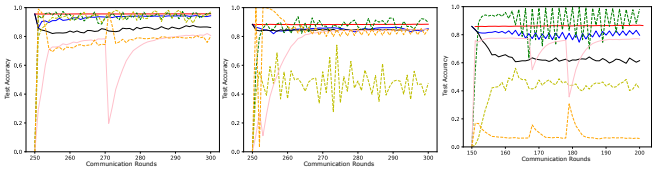
Fig. 3 The test accuracy changes with the number of virtual nodes generated by each malicious clients while fix the proportion of malicious clients to 0.4.

The results in the figures show that, across all three datasets, increasing either $m\%$ and v improves the attackers target task precision on the test dataset. Notably, increasing the proportion of malicious clients has a more immediate impact on attack performance, as it provides more client data for computing the target model. However, both increasing the proportion of malicious clients and increasing the number of sybil nodes generated by each malicious client reduce the accuracy of the main task, although the overall decrease is modest. Thus, balancing target task accuracy and main task accuracy, we set $m\% = 40\%$ and $v = 5$ for subsequent experiments.

C. Comparative Experiment

We implement the method from [19] within our sybil-based data poisoning framework. In this methode, malicious clients generate poisoning data using the global model and sends it to the sybil nodes for poisoning. This method is referred as the Feature Collision Method (FCM). The approach in [20] is similar to the online local target model scheme we proposed, with the key difference being that it only considers the target class of images in the counter loss, excluding benign samples from other classes. We refer to this method as the Local Method (LM).

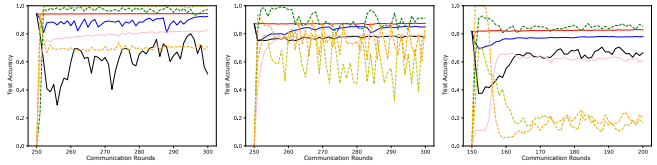
Non-independent, identically distributed (Non-IID) data is a key characteristic of federated learning, and varying heterogeneous data distributions can influence the effectiveness of poisoning attacks. Using the two methods described above and our proposed method, we conducted comparative experiments with three data distributions: IID, Dirichlet distribution with $\alpha = 0.5$, and Dirichlet distribution with $\alpha = 0.1$. The experimental results are shown in Fig.4, Fig.5, and Fig.6.



(a) MNIST dataset **(b)** FMNIST dataset **(c)** CIFAR-10 dataset

GMA(without malicious client) MTA of our method TTA of our method MTA of LM
TTA of LM MTA of FCM TTA of FCM

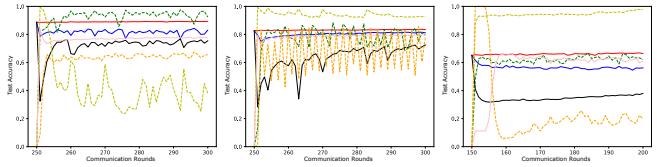
Fig. 4 Attack performance when three datasets are IID.



(a) MNIST dataset **(b)** FMNIST dataset **(c)** CIFAR-10 dataset

GMA(without malicious client) MTA of our method TTA of our method MTA of LM
TTA of LM MTA of FCM TTA of FCM

Fig. 5 Attack performance when the Dirichlet distribution has parameter α = 0.5.



(a) MNIST dataset **(b)** FMNIST dataset **(c)** CIFAR-10 dataset

GMA(without malicious client) MTA of our method TTA of our method MTA of LM
TTA of LM MTA of FCM TTA of FCM

Fig. 6 Attack performance when the Dirichlet distribution has parameter α = 0.1.

Overall, the method proposed in this paper outperforms others across all three datasets in both main task accuracy (MTA) and target task accuracy (TTA). Specifically, the poisoning data we generate is highly dependent on the target model, and its quality is influenced by the dataset held by the controlled malicious clients. Even when using the Dirichlet distribution with $\alpha = 0.1$, which simulates highly non-independent data, the poisoning attack remains effective. On these three datasets, the TTA of the proposed method is 92.42%, 80.43%, and 63.74%, respectively. Although the TTA is not the highestbeing 12.27% and 33.96% lower than the LM method for FMNIST and CIFAR-10the MTA of LM has significantly decreased, rendering the poisoning attack less effective. The experimental results show that the data poisoning attack method proposed in this paper plays a significant role in improving the performance of the poisoning attack by considering non-target samples in the objective function and obtaining the global target model on the non-IID data.

V. CONCLUSION

In this paper, we have proposed a sybil-based virtual data poisoning attack that leverages sybil nodes to amplify the impact of the poisoning attack while minimizing the high cost associated with malicious clients directly providing data. Our method first computes the target model, then generates virtual data on the malicious client, which is distributed to the corresponding sybil node for federated learning participation, thereby poisoning the global model. In experiments, we explore the appropriate proportion of malicious clients and the number of sybil nodes using non-IID datasets. We have also compared our approach with existing algorithms under various data distributions. Compared to the second-best local method, the main task accuracy improved by 7.6%, 9.03%, and 17.3%.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[2] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017.

[3] R. Wang, B. Fu, G. Fu, and M. Wang, "Deep & cross network for ad click predictions," in *Proceedings of the ADKDD'17*. New York, NY, USA: Association for Computing Machinery, 2017, pp. 1–7.

[4] Q. Guan, W. Li, S. Xue, and D. Li, "High-resolution representation object pose estimation from monocular images," in *2021 China Automation Congress (CAC)*. IEEE, 2021, pp. 980–984.

[5] Q. Guan, Z. Sheng, and S. Xue, "Hrpose: Real-time high-resolution 6d pose estimation network using knowledge distillation," *Chinese Journal of Electronics*, vol. 32, no. 1, pp. 189–198, 2023.

[6] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 654–17 665, 2022.

[7] Z. Sheng, L. Liu, S. Xue, D. Zhao, M. Jiang, and D. Li, "A cooperation-aware lane change method for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 3236–3251, 2022.

[8] Y. Guan, D. Li, S. Xue, and Y. Xi, "Feature-fusion-kernel-based gaussian process model for probabilistic long-term load forecasting," *Neurocomputing*, vol. 426, pp. 174–184, 2020.

[9] S. Jia, Z. Gan, Y. Xi, D. Li, S. Xue, and L. Wang, "A deep reinforcement learning bidding algorithm on electricity market," *Journal of Thermal Science*, vol. 29, no. 5, pp. 1125–1134, 2020.

[10] L. Liu, J. Zhang, and S. Xue, "Photovoltaic power forecasting: Using wavelet threshold denoising combined with vmd," *Renewable Energy*, vol. 249, p. 123152, 2025.

[11] L. Lyu, Y. W. Law, K. S. Ng, S. Xue, J. Zhao, M. Yang, and L. Liu, "Towards distributed privacy-preserving prediction," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 4179–4184.

[12] 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*, 2017, pp. 1273–1282.

[13] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[14] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021.

[15] Q. Wu, L. Liu, and S. Xue, "Global update guided federated learning," in *2022 41st Chinese Control Conference (CCC)*. IEEE, 2022, pp. 2434–2439.

[16] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv:1206.6389*, 2012.

[17] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *25th European symposium on research in computer security*, 2020, pp. 480–501.

[18] V. Shejwalkar, A. Houmansadr, P. Kairouz, and D. Ramage, "Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2022, pp. 1354–1371.

[19] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," *Advances in neural information processing systems*, vol. 31, pp. 6106–6116, 2018.

[20] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, "Witches' brew: Industrial scale data poisoning via gradient matching," *arXiv:2009.02276*, 2020.

[21] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, 2020, pp. 2938–2948.

[22] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International conference on machine learning*, 2019, pp. 634–643.

[23] X. Zhou, M. Xu, Y. Wu, and N. Zheng, "Deep model poisoning attack on federated learning," *Future Internet*, vol. 13, no. 3, pp. 73–88, 2021.

[24] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, 2020, pp. 301–316.

[25] X. Xiao, Z. Tang, C. Li, B. Xiao, and K. Li, "Sca: Sybil-based collusion attacks of iiot data poisoning in federated learning," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 2608–2618, 2022.

[26] X. Xiao, Z. Tang, C. Li, B. Jiang, and K. Li, "Sbpa: Sybil-based backdoor poisoning attacks for distributed big data in aiot-based federated learning system," *IEEE Transactions on Big Data*, vol. 10, no. 6, pp. 827–838, 2024.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

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

[29] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto*, pp. 1–58, 2009.

[30] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.