Targeted Adversarial Poisoning Attack Against Robust Aggregation in Federated Learning for Smart Grids

Guihai Zhang and Biplab Sikdar, *Fellow, IEEE*

***Abstract*—Federated learning has been widely adopted as an intelligent computing method to leverage the power of distributed data without compromising data privacy and security, especially in smart grids. Poisoning attack is one of the common attacks against federated learning to degrade the performance of the global model or cause it to make specific, incorrect predictions. To counter these threats, secure aggregation rules have been implemented to reduce the impact of adversarial or malicious updates during training process. In this paper, we first propose a norm-based aggregation rule specifically designed to mitigate the effects of poisoning attacks within federated learning systems used for power quality classification. Subsequently, we introduce the Targeted Adversarial Poisoning Attack (TAPA), a method that aims to break the secure aggregation rule by causing the global model to misclassify a specific targeted class as another chosen class with increased success. This attack method employs a generative approach to produce extra fake data and trains a malicious model whose parameter norms closely resemble those of standard local models, thereby increasing the likelihood of these updates surviving the filtering processes of secure aggregation rules. Evaluation results demonstrate that the proposed TAPA method can achieve a higher poisoning attack success rate while preserving the classification accuracies of non-targeted classes, even in the presence of unbalanced data distributions. This approach underscores the potential vulnerabilities in existing secure aggregation mechanisms and highlights the need for robust defenses against sophisticated poisoning strategies in federated learning environments.**

***Index Terms*—Power quality classification, federated learning, poisoning attack, label flipping, robust aggregation rule, adversarial attack.**

I. Introduction

FEDERATED learning arises in response to the increasing attention to the private and sensitive information that are contained in available datasets for data-driven applications in smart grids. It is a distributed approach that allows multiple participants to collaboratively train a shared model while keeping their data decentralized and private [1]. This framework aligns well with various smart grid applications, such as anomaly detection, intrusion detection, and energy prediction [2], [3], [4]. Smart sensors and meters independently collect their local data, eliminating the need to transmit this data to a central server, thereby maintaining data privacy. For example, in [5], a federated learning application is designed to predict electricity consumption based on the collaborations of smart meters. In this context, every household is equipped with a smart meter that collects high-resolution electricity usage data. This data is highly sensitive, as it can reveal detailed patterns of household behavior, occupancy, and appliance usage. Therefore, sharing raw consumption data with a central server raises serious privacy concerns, especially under protection regulations. Federated learning provides a privacy-preserving solution by enabling smart meters to collaboratively train a global load forecasting model without sharing their raw data. Instead, each smart meter locally trains a model on its own electricity usage data and only shares model updates (such as weights or gradients) with a central aggregator. These updates are then aggregated using an algorithm to improve a shared global model. In a typical federated learning process as shown in Fig. 1, a global model is initialized on a central server and sent to participating nodes and each node uses its local data to train the model for multiple iterations, producing local updates. The central server receives and processes the local model updates to refine the global model where the aggregation process can range from simple averaging of the updates to more sophisticated methods incorporating defense strategies designed to mitigate the impact of malicious behavior. The updated global model is redistributed to all

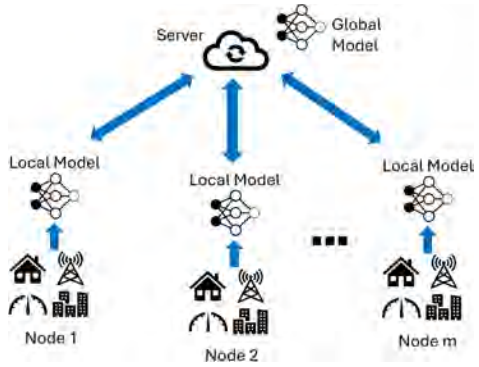

Fig. 1. Federated Learning Framework in Smart Grid.

Received 8 October 2024; revised 23 July 2025; accepted 31 October 2025. Date of publication 5 November 2025; date of current version 23 February 2026. This work was supported in part by the Agency for Science, Technology and Research (A*STAR), CISCO Systems (USA) Pte. Ltd.; and in part by the National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory under Award I21001E0002. Paper no. TSG-01730-2024. *(Corresponding author: Guihai Zhang.)*

The authors are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 119077 (e-mail: e0534010@u.nus.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2025.3629243.

nodes for continuous training. This cycle is repeated until the model converges to a satisfactory performance level. The nodes involved in this process can be individual devices or clusters of neighboring devices. Importantly, during training, local data remains confined to the originating node, with no direct data exchange between nodes or with the central server, thereby ensuring that data privacy is maintained throughout the distributed training process.

However, there are potential security and privacy risks in federated learning such as poisoning attacks and privacy leakages [6], [7]. Poisoning attacks are a form of adversarial attack where an attacker seeks to corrupt the learning process by injecting malicious data or model updates into the system [8]. These attacks are designed to either impair the performance of the global model or induce it to produce specific, incorrect predictions. Successful attacks against smart grid federated learning systems could lead to consequences like wrong power flow analysis, inaccurate energy prediction, imbalanced power supply and demands, and even blackouts, if crucial equipment is disrupted [9]. Thus, there are also defense strategies and techniques to ensure the integrity and robustness of the global model despite the attacks from malicious participants. Such defense methods include robust aggregation rules, model analysis methods, and verification-based methods [8], [10], [11]. These defense methods can be implemented individually or in combination to detect and mitigate poisoning attacks. They seek to ensure that federated learning systems remain reliable in adversarial environments, safeguarding both the integrity of the models and the interests of the participants.

In this paper we first propose a simple yet effective norm-based aggregation rule as a defense method against known poisoning attacks. This aggregation rule evaluates the L_2 norm of each local model update and discards the largest and smallest outliers. Only the remaining local model updates are averaged to update the global model. Building on the analysis of this secure aggregation rule's performance, we further introduce the Targeted Adversarial Poisoning Attack (TAPA) to perform a targeted label flipping attack that can bypass the secure aggregation rule. The goal of the attack is to undermine the global model to incorrectly classify the targeted label to a chosen new label, by introducing malicious model updates to the global model at one or few local nodes, without accuracy reduction in other labels. The proposed attack method contains two main parts: poison data preparation and malicious parameters constraint. The poison data preparation phase allows the attacker to approximate the patterns of global local data by generating fake data for all labels using a generative model. The malicious parameter constraint aims to ensure that the parameters of the crafted malicious model closely resemble those of benign models, thereby increasing the likelihood of the malicious model's survival during robust aggregation.[1] This is achieved by minimizing the difference in magnitudes between the malicious and approximated global model parameters. The key contributions of the paper are as follows:

- We introduce a simple robust aggregation rule by removing outlier model parameters that have largest and smallest L_2 norm. The method is examined on a typical existing poisoning attack.
- We propose the TAPA to conduct targeted label flipping attack against federated learning in power system applications, and it is a general method under both independent and identically distributed (IID) and non-IID situations.
- We refine the malicious model training process by considering the matching between model updates from benign and malicious models to increase the malicious parameters' survival rate in robust aggregation rule.
- We evaluate the proposed TAPA with existing similar work and show that TAPA has better performance in both poisoning attack success rate and survival rate in robust aggregation rule, especially with less compromised local models.

TABLE I
KEY TERMINOLOGIES

Term	Definition
Local model	A copy of the global model maintained and trained independently by each local client (e.g., smart meter). It is initialized with the current global model received from the central server and updated using the client's local dataset.
Global model	The central model maintained by the server that is periodically updated by aggregating local updates from distributed clients. This model is initialized and controlled by the central server and sent to local clients at the beginning of each training round.
Local updates	Model weight updates computed by individual clients (e.g., smart meters) during local training using local private data.
Global update	The aggregated model obtained by the central server by combining local updates from multiple clients to form the updated global model.
Targeted label	The specific class label that an adversary aims to cause benign inputs to be misclassified as, in a targeted poisoning attack.
Label flipping	A strategy where an adversary alters the class labels in local training data (e.g., flipping class A to B) to inject misclassifications.
Aggregation rule	The strategy used by the server to combine local updates.

The key terminologies used in this paper are summarized in Table I.

The subsequent sections of the paper are as follows. Section II provides the literature review of related works. Section III describes the proposed method in detail. Section IV shows the settings of evaluation. Section V presents the results and discusses the methods. Finally, Section VI concludes our work.

[1] A malicious model is a local model owned by attacker/malicious node and is used to poison the global server model by training with poisoned dataset. A benign model refers to a model maintained by a benign node, which is updated through standard training processes using a non-malicious local dataset.

II. Related Works

A. Poisoning Attacks

Poisoning attacks can be divided into two categories: data poisoning attacks and model poisoning attacks. In a data poisoning attack, attackers can only modify the local data. This attack is done by falsifying and tampering with the local data, like changing the labels of data or adding random noise into data [12]. Since these methods cannot modify the model parameters directly, the attack success rate is limited. Model poisoning attacks enhance their effectiveness by carefully manipulating the local model parameters before sending them to the server, thereby influencing the updates of the global model. The work in [13] uses a random initial model with low accuracy as a base model and drags the global model towards the base model in each training round. Moreover, [14] performs untargeted model poison attacks by getting benign updates from benign local data and adding perturbations to deviate the global model. While these methods require attackers to compromise benign local nodes first, there are other methods that can obtain malicious model updates without access to other benign data. The work in [15] uses public interactions to approximate the users' feature vectors. PoisonGAN in [16] uses a generative network to generate fake samples and change the label of targeted data to a chosen label to form the poisoned dataset. The malicious model is trained on the poisoned dataset and the model updates are further scaled up to amplify the effects of the malicious model. Similar work that uses Generative Adversarial Network (GAN) to generate specific samples to corrupt a targeted intrusion detection model is done in [17]. It is noteworthy that many methods necessitate a scaling factor to enhance the impact of the malicious model when aggregated at the server. This scaling is employed to mitigate the reduction in effectiveness that occurs when the malicious model is averaged with other local updates. This reliance on scaling factors represents a vulnerability that can be exploited by defense mechanisms, especially when malicious and normal nodes have similar data distribution. There are other works to conduct manipulation on specific weights of a model to perform more subtle poisoning strategies. Specifically, Fang's method in [18] uses a strategy to compute the mean and standard deviations of the poisoned models's weights and sample from the 95% tail of each side of the distribution. This method introduces an effective way to handle limitations of no knowledge of other benign updates. However, this work is a non-targeted poisoning attack and it requires a sufficient number of compromised models. Further evaluations should have been done to examine the effectiveness of this method to perform targeted attack, and in scenarios with various different number of attacker-owned models.

B. Byzantine-Robust Aggregation Rules

Malicious updates from federated learning participants can be identified and removed using Byzantine-robust aggregation rules. The rules are applied to examine characteristics of received model updates and to find any abnormal updates that are different from majority of all received updates. Only those updates that pass specific rules are used for the final aggregation of the global model. Krum or Multi-Krum [19] selects the update that is most similar to the majority by computing scores based on the sum of distances to the nearest updates from other participants. These methods aim to identify suspicious parameters in models and exclude those models from global updating. There are other methods that focus on individual parameter. The trimmed mean or median method in [20] initially removes b largest and b smallest values in each parameter from m local models. Then, the global model parameter is updated based on the mean or median parameter values of the remaining $m - 2b$ local updates. It is noted that those trimmed mean or median methods ingest the updates from all models including the malicious updates. The trimming process is done on the individually,per-parameter. This introduces the potential risk of neglecting the joint statistical correlation across malicious parameters. Adversaries are able to manipulate multiple parameters in a coordinated fashion while staying within the acceptable per-parameter ranges. Consequently, such methods may fail to detect well-crafted poisoning updates that are subtle in each dimension but collectively harmful.

Gradient clipping applies a bound to the norm of local updates in order to eliminate the effects of poisoning attacks [21]. However, the performance of this method highly depends on the choice of the bound threshold. Another method is proposed in [18] where the server has a validation dataset to compute the error rate of each local update. However, this method's reliance on a validation dataset at the server side may not always be feasible or practical, as such datasets may not be available in all scenarios.

Based on the reviews of existing methods, the PoisonGAN method in [16] is the most similar work to ours that performs the targeted poisoning attack. It compromises one or few local nodes and trains the malicious model on generated poisoned dataset. Despite its novelty, the performance of the attack needs to be evaluated when facing a robust and defensive aggregation rule. Therefore in this paper, first a simple yet robust norm-based aggregation rule is proposed to mitigate the impacts of this existing targeted attack. Next, the paper proposes a new attack strategy with improved poisoned data generation and malicious model training processes. For a more comprehensive analysis, the exiting trimmed mean aggregation rule and Fang's attack method are also applied in this paper to highlight the proposed method's advantages and limitations. It is noted that there is no existing poisoning attack exactly the same as ours; Fang's method is adapted to perform targeted attacks despite its original goal of untargeted model degradation. This method is used as a strong baseline due to its design, which enables the attacker to craft updates that are both stealthy and effective under robust aggregation rules.

III. Methodology

A. Threat Model and Assumptions

In this paper, we focus on a malicious scenario specifically targeting a power quality classification application. In this scenario, attackers can compromise a certain number of local

nodes and upload malicious model parameters to the server. When those malicious parameters are aggregated with other local updates, the global model will learn from the malicious models, resulting in the classification of targeted data as an attacker-chosen label. This entire process is referred to as a targeted poisoning attack. We define the training process of the malicious model as malicious training and the dataset used by this model is the poisoned dataset. The detailed assumptions and settings are as follows:

- In the distributed power quality classification federated learning application, there are multiple local data centers/nodes to collect power data locally within a region. Each local node has an intelligent device to train a local model on-site using its power quality measurements.
- Since the global model is initialized by the central server and will be delivered to each local node, the attacker is able to obtain the latest status of the global model.
- The attacker has the ability to compromise a fixed number of local data centers/nodes by malware injection via phishing or insecure software supply chains, credential theft leading to unauthorized remote access, etc.
- These compromised nodes contain the original data which attackers could exploit to perform malicious training using poisoned or synthetically generated data, and upload manipulated model updates to the server.

B. Norm-Based Aggregation Rule

The approach is inspired by the trimmed mean method [20] but simplifies the process by sorting the $L2$ norm, rather than comparing each individual parameter. Poisoning attacks that employ scaling factors to upload malicious model parameters to the server can significantly increase the norms of these parameters, making them conspicuous among the received parameters. Therefore, it is reasonable to exclude outlier parameters with the largest norms, as these are more likely to be associated with malicious models. In addition, the parameters with very small norms might indicate uninformative or poorly trained updates, which could be the result of nodes that did not train their models effectively, are malfunctioning, or are adversarial. Thus, the model parameters with the smallest norm are also removed during aggregation. Parameters with very small norms often indicate that the corresponding local models contributed minimal information to the global learning process. There are several reasons for this. Firstly, if a node has insufficient data, poor data quality, or fails to complete local training properly (due to low computation, memory issues, or early interruption), its resulting weight updates will exhibit very small magnitudes. Secondly, malfunctioning nodes may return near-zero or default values due to internal errors or software faults, again leading to negligible update norms. Lastly, in adversarial settings, malicious clients might deliberately suppress their gradients or mimic undertrained behavior to evade detection or confuse the aggregation mechanism. In all these cases, the small norm becomes an indicator of an uninformative or potentially harmful update, justifying its reduced influence or removal during robust aggregation. The model parameter after local training is denoted as w_i for the i-th node. Then, the $L2$ norm (or magnitude of the parameter vector) is:

$$M_i = \|w_i\|_2 = \sqrt{\sum_j w_{i,j}^2}, \tag{1}$$

where j represents the j-th element of the parameter vector w_i. Then, for a total of N received local model parameters, $[M_1, M_2, \ldots, M_n]$ is the list of the magnitudes of all the local models. The next step is to find the largest and smallest values in all the magnitudes and drop the corresponding model parameters. The average parameter w_a is obtained by computing the average value of all the remaining surviving models:

$$w_a = \frac{1}{N-2} \sum_{k \in T} w_k, \tag{2}$$

where T is the set of indices of the remaining models. Thus, the global model parameter w_g^{t+1} is obtained by:

$$w_g^{t+1} = (1-\alpha) \cdot w_g^t + \alpha \cdot w_a. \tag{3}$$

Here, α is the global model learning rate between 0 and 1 to determine the degree of contribution of the latest averaged parameter vector to the global model parameters. Although this rule also gets the norms of the model parameters, it is simpler than the popular [21], since there is no effort needed to find the optimal threshold value. The proposed aggregation rule helps in maintaining a robust model by focusing on updates that are more in line with the majority.

The difference between this method and existing trimmed mean method is the collection of malicious models. The trimmed mean method evaluates all parameter values across all client models, regardless of whether a model is benign or malicious. It then removes the largest and smallest values for each parameter independently across the models. This per-parameter trimming disregards the structural coherence of each client's update, allowing adversarial updates to bypass detection by distributing malicious perturbations across multiple dimensions in a coordinated but subtle way. In contrast, our method takes into account the model-level consistency and adversarial intent, aiming to expose malicious behavior that might be masked in dimension-wise statistics alone. The model with extreme values of parameters is discarded as a whole.

C. Targeted Adversarial Poisoning Attack

The operation of the proposed TAPA method is shown in Fig. 2. The attacker owns a certain number of local nodes and has the corresponding local dataset. Then, the labels of targeted data are changed to a chosen new label to form the poisoned dataset (e.g., change all "1" to "4"). The malicious model is trained on the poisoned dataset so that this model will classify the targeted class data as the chosen label (e.g., classify class 1 data to be class 4 data). The parameters of the malicious model are sent to the server and aggregated with the other local model parameters. It is expected that with the continuous uploading of the malicious model parameters, the global model will be influenced and have similar behavior as the malicious model. The ultimate outcome as indicated by the black arrow in Fig. 2 is to let the global model also classify

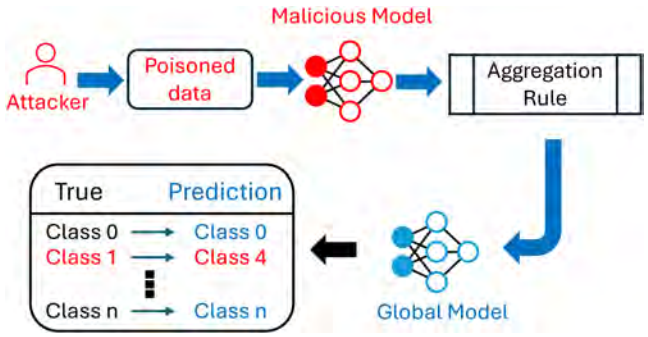

Fig. 2. Illustration of the Targeted Adversarial Poisoning Attack flow.

the targeted class data (class 1) as the chosen label (class 4). The attacker should achieve two objectives at each federated learning round: poisoned data preparation and malicious model training.

1) Poisoned Data Preparation: The first goal is to form the poisoned dataset. There may be situations where the local dataset does not contain the targeted data. This could be solved by using the generative method in [16]. This method employs a GAN, where a generator neural network creates synthetic samples and a discriminator classifies them. We propose to use the global model as the discriminator. This is feasible because the global model is inherently a classification model and it maps input data to class labels. Moreover, it is updated through the aggregation of local model parameters, reflecting knowledge from all clients. By substituting the discriminator with the global model (i.e., sharing both its structure and parameters), we enable the generator to produce data that is recognizable by the current global decision boundaries. The generator G is a neural network that receives a random noise vector z and generates fake samples $x' = G(z)$. During each local training round at an attacker node, the following steps are performed:

- Discriminator Initialization: The attacker downloads the latest global model and uses it as a fixed discriminator D. No separate discriminator model is trained.
- Enumerate the lables y from all available Y.
- Data Generation: The generator produces synthetic samples from noise input.
- The discriminator D assigns labels $y' = D(G(z))$ with highest probability to these samples.
- Loss Computation: A classification loss is computed between the predicted label y' and current enumerated label y.
- Generator Update: The generator is updated to minimize the following loss:

$$L_G(\theta_g) = \mathbb{E}\left[\log(D(G(z)))\right]. \tag{4}$$

  This objective encourages the generator to produce examples that the global model confidently classifies into the desired class.

This process is repeated for several iterations, enabling the generator to gradually learn how to create samples that match the target classes from the perspective of the global model. Since the discriminator is not explicitly trained and only the generator is updated, no additional real data is needed for training either component. All supervision comes from the global model's classification outputs, making this process fully self-contained and data-independent on the attacker side. As a result, the attacker can generate a complete synthetic dataset covering all target classes. This generative poisoning strategy is lightweight, requires no access to real data, and remains compatible with the federated learning pipeline. The structure of the generator used in this paper is shown in Table II. There are batch-normalization and leaky-relu processes between each layer.

TABLE II
STRUCTURE OF GENERATOR

Layer	Details
1	Dense(512)
2	Conv1DTranspose(256)
3	Conv1DTranspose(128)
4	Conv1DTranspose(128)
5	Flatten()

Algorithm 1 Fake Sample Generation

Require: G, D, Y, NUM
for $y \in Y$ **do**
  while $num \leq NUM$ **do**
    $z \sim \mathcal{N}(\mu, \sigma^2)$
    $x' = G(z)$
    if $D(x') == y$ **then**
      Add x' to fake dataset D_f, with label y.
    end if
    Update G based on (4).
  end while
end for
return fake dataset D_f

However, the generation of fake samples in this paper differs from the approach used in [16]. Fake data of all classes are generated at the attacker node, instead of generating only targeted class data. This allows the attacker to train a malicious model that has general knowledge of patterns across all data. The original method in [16] only uses the generated targeted data to train the malicious model. There is only one single class data trained in a model. Thus, the model's effectiveness is limited and it does not learn the underlying characteristics of the poisoned data. Instead, the model merely outputs the assigned single label without effectively capturing the nuances or patterns inherent in the data. Thus, it is crucial to let the malicious model to accurately learn the underlying patterns of the targeted data. Additionally, generating fake samples across all classes supports the malicious training process by enabling the creation of a malicious model that exhibits comparable patterns to that of a benign local model. The process of fake sample generation is shown in Algorithm 1. NUM defines the maximum number of data of each class and D_f contains the fake data belonging to all classes. Then, the poisoned dataset D_p is obtained by combining the real local data and the generated fake data D_f, and the labels of the targeted

data have to be changed to the attacker-chosen new label. This process is the data augmentation that not only keeps the original local data information but also introduces the extra poisoned information. The malicious model can learn the benign patterns and the poisoned patterns at the same time. In this paper, the *NUM* is set to be 500 to get 500 data points of each class.

2) Malicious Model Training: The second essential part is to train a malicious model that will learn from the patterns of the poisoned dataset and survive in a robust aggregation rule to influence the global model. The proposed norm-based aggregation rule and the method in [21] are analyzed to provide insights into their effectiveness in mitigating poisoning attacks. Moreover, it is noted that all existing poisoning attack methods that amplify the model parameters to gain large impacts on the global model make them distinguished from the benign local models. Thus, there is a trade-off between the poisoning effects and the survival rate under robust aggregation rules. Thus, a new malicious model training method is proposed to constrain the norm of a malicious model to increase the survival rate, without diminishing its capacity to poison the global model. It is assumed that the final global model after federated learning should have learned all the patterns of all classes of data in a situation where every node is benign. Then it is reasonable to allow the malicious model to also acquire knowledge of these benign data patterns, making it more similar to the global model. This similarity by aligning malicious updates with the patterns of the benign model can make it challenging to detect the malicious model's influence, as its updates can appear more consistent with the overall benign data distribution. Thus, the malicious model updates can blend seamlessly with those from legitimate local models at each federated learning round. The challenge for the attacker node is how to get all the benign local data to learn their patterns. It is easy if the compromised node has data of all classes but there are situations where the node only has a certain part of the whole data distribution. This can be solved by using the generated fake samples D_f as described in the last section.

Algorithm 2 Malicious Model Training

Require: w_g, K
1: Get D_f and D_p by using Algorithm 1.
2: Benign model f_b: $w = w_g$
3: Train benign model on D_f.
4: Get benign norm M_b.
5: Malicious model f_m: $w_m = w_g$
6: **while** $k \leq K$ **do**
7: Get standard L_{CE} on D_p.
8: Get malicious norm M_m.
9: Get malicious loss L_{norm}.
10: Get total loss $L_T = L_{CE} + L_{norm}$.
11: Update the malicious model.
12: **end while**
13: **return** Malicious model f_m.

At the beginning of local model training process, the attacker uses the generator to get the fake samples D_f and poisoned data D_p. A benign local model f_b is initially trained on the fake samples D_f to get the benign model parameters w_b. Next, the malicious model f_m with parameters w_m is trained on the poisoned data D_p. The loss function of the malicious model includes the combination of standard categorical Cross Entropy loss and the norm-matching loss. The Cross Entropy loss is:

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i), \tag{5}$$

where y is the true label and $\hat{y}$ is the prediction. Another loss is the norm-matching loss, which is introduced to constrain the model parameters. This represents the absolute difference between the norm M_b of benign model parameters and the norm M_m of malicious model parameters. It can be expressed as:

$$L_{norm} = |M_b - M_m|, \tag{6}$$

where norm M is calculated from Equation (1) and L_{norm} denotes the loss function quantifying the discrepancy between the norms of the benign and malicious parameters. Thus, the overall loss function for the malicious model to optimize is:

$$L_T = L_{CE} + L_{norm}. \tag{7}$$

The training process of the malicious model at each federated learning round is shown in Algorithm 2. The global model parameters w_g are set for all the local models and the constant K defines the number of local malicious training steps. It can be seen that there is no scaling factor applied to the malicious model parameters. The task of label flipping is accomplished by training the malicious model on a poisoned dataset. This dataset is augmented with both genuine local data and artificially generated fake data, allowing the malicious model to learn the typical patterns of benign data. Thus, the malicious model parameters can survive with higher probabilities when aggregated with other local updates. Consequently, the model can not only align to these normal patterns but also classify targeted data to the attacker-chosen new label.

While the proposed TAPA method can address the limitation of absence of poisoned data, the existing Fang's method does not consider this. Since Fang's method is an untargeted attack and it tries to misclassify any label of the prediction, it pushes the poisoned model to the opposite direction of gradient update. Moreover, Fang's method requires some amount of compromised local nodes to compute the mean and standard deviation of malicious parameters. Then, the final poisoned parameter is sampled from the obtained distribution for each parameter. This method has the limitation that less compromised local models will reduce the precision of the calculation. In addition, this assumption of distribution made according to the unknown benign updates may not always be true. The proposed TAPA method is able to address these concerns with the generation of fake benign data. The poisoned model is trained to have targeted attack goal and to follow certain underlying patterns of normal updates as well.

IV. Evaluation

An initial evaluation is conducted on proposed TAPA and PoisonGAN methods against proposed norm based aggregation rule to check the effectiveness of targeted poisoning

TABLE III

SETTINGS OF FEDERATED LEARNING SYSTEM WITH NORM-BASED AGGREGATION RULE

Configurable Parameter	Value
No. of local nodes (P)	10
No. of selection (S)	6, 8, 10
Local training steps	10
Local learning rate	0.0001
No. of largest norm dropping	1
No. of smallest norm dropping	1
No. of aggregating local updates	4, 6, 8
Global learning rate	0.1

TABLE IV

STRUCTURE OF GLOBAL CLASSIFICATION MODEL

Layer	Details
1	Dense(256)
2	Dense(128)
3	Dense(16)
4	Dropout(0.2)
5	Dense(5)

attack. To have further comprehensive analysis, comparisons are made by implementing Fang's poisoning attack and trimmed mean aggregation rule. Fang's method is well-known for its ability to significantly degrade global model performance under trimmed mean, making it a useful reference point for understanding the relative strength of targeted attacks. Although originally designed for untargeted poisoning, Fang's method can be adapted to perform targeted attacks. This is achieved by adjusting the direction of parameter perturbations toward a specific target class and incorporating our proposed fake data generation strategy in this evaluation. By integrating these modifications, we are able to align Fang's framework with the goals of targeted poisoning, enabling a fairer comparison with TAPA and PoisonGAN.

A. Federated Learning Configuration

To evaluate the norm-based aggregation rule and the proposed TAPA, a federated learning system of training a power quality classification model is constructed. The details of the federated learning setting with the proposed norm-based aggregation rule are summarized in Table III. The total number of local nodes (P) is fixed at 10. However, the number of participating nodes (S) can vary to facilitate the analysis of poisoning attacks under different scenarios. In the context of the norm-based aggregation rule, the number of verified local model updates is set to $N-2$, resulting in 8, 6, and 4 updates, respectively. The scalar α in Equation (3) is set to 0.1. The global classification model to be trained has the structure shown in Table IV. The model has 66,565 parameters, indicating that it is relatively lightweight.

The dataset is from the Amrita Honeywell Hackathon 2021 and it contains measured voltage signals [22]. Each signal is featured by 128 sampling points from two sampling cycles and the fundamental frequency is in the range of 59.5 and 60.5Hz. All signals come from five categories: normal (label 0), 3-rd harmonic wave (Third-h: label 1), 5-th harmonic wave (Fifth-h: label 2), voltage dip (label 3) and transient (label 4). They are divided into training (9598) and testing datasets (2400). In order to obtain a more practical simulation under federated learning, the training dataset is divided randomly to each node and is non-IID. The assignment is shown in Table V. The total number of data points of each class is different and the total number of data points on each node is different. Various scenarios have been considered, including cases where all labels are present with comparable counts, cases where all labels are present but some are in the majority, cases with an abundance of certain labels and scarcity of others, and cases with an absence of normal labels but the presence of other specific labels. This approach allows for testing in a more realistic and general setting, reflecting a broader range of possible data distributions. It also makes this paper more comprehensive than [16] where the authors only run evaluations on evenly distributed data. The skewness of the data distribution is summarized in Table VI. The Jensen-Shannon Divergence (JSD_to_Global) measures how different each group's data distribution is compared to the global distribution. Higher values imply that the group has a more skewed or unique distribution compared to the global dataset. It is shown that the local nodes have covered all possible scenarios of the data distribution. For example node 1 has the lowest JSD_to_Global value. It shows that this node has the data distribution that is the most similar to global data distribution. Then, nodes 5, 8 and 9 have higher values to show that their local data distributions are different from global distribution. The next metric is the Entropy. It measures the diversity or uniformity within the group's class distribution. Higher entropy means more balanced class distribution in the group. The local nodes have the values ranging from 0.9237 to

TABLE V

NUMBER OF DATA POINTS IN EACH NODE

Node	Class 0	Class 1	Class 2	Class 3	Class 4	Sum
0	300	129	530	200	160	1319
1	200	200	300	200	235	1135
2	150	180	410	125	380	1245
3	200	100	145	0	555	1000
4	310	180	335	195	0	1020
5	249	0	0	120	230	599
6	190	0	150	0	200	540
7	0	220	550	240	100	1110
8	0	400	0	205	0	605
9	0	200	0	300	525	1025
Sum	1599	1609	2420	1585	2385	9598

TABLE VI

DISTRIBUTION SKEWNESS OF DATA POINTS

Node	JSD to Global	Entropy	Imbalance Ratio
0	0.0389	2.1243	4.1085
1	0.0018	2.3018	1.5000
2	0.0149	2.1545	3.2800
3	0.1437	1.6720	5.5500
4	0.1433	1.9477	1.8611
5	0.2614	1.5214	2.0750
6	0.2000	1.5743	1.3333
7	0.1436	1.7553	5.5000
8	0.4690	0.9237	1.9512
9	0.2568	1.4732	2.6250

TABLE VII
EXPECTED NUMBER OF ATTACKER NODES SELECTED UNDER EACH FEDERATED LEARNING SETTING

Number of A	ratio of S / P		
	6/10	8/10	10/10
1	1	1	1
2	1	2	2
3	2	2	3

2.3018. To be noted, the high entropy value near 2.32 (since $\log_2(5) \approx 2.32$) shows the local class distribution is nearly IID. Thus, there is local node 1 that has local data with IID distribution (entropy value of 2.3018). The other nodes like 7 and 8 have less balanced class distribution. The imbalance ratio measures how imbalanced the non-zero class counts are within a group. Value of 1 means that the data is perfectly balanced among non-zero classes. The larger the value, the more skewed the local distribution (e.g., one class dominates the others). It can be seen that there are most balanced nodes: node 1, 4 and 6; moderately imbalanced nodes: node 0, 2, 5 and 9; and most imbalanced nodes: node 3 and 7. These three metrics together provide a comprehensive view of the non-IID nature of the local datasets.

B. Poisoning Attack Configuration

The poisoning attack is evaluated under both single-node and multi-node attack scenarios. In the single-node attack scenario, the adversary conducts the poisoning attack by compromising a single node. Conversely, in the multi-node attack scenario, the adversary carries out the attack by compromising multiple nodes. Specifically, in this poisoning attack, the label of class "1" is altered to label "4" (i.e., Third-h to Transient). Furthermore, an evaluation of flipping label class "4" to label "1" is also conducted to provide an analysis of the effectiveness of proposed method under a different scenario. This allows evaluation under both cases: flip labels from a smaller group to a larger group, and flip labels from a larger group to a smaller group. To perform the attack, the attacker can compromise a number A of local nodes. The local nodes owned by the attacker are node 1, 5 and 7. This selection encompasses all representative cases: nodes with data from all classes, nodes without the targeted data, and nodes without the normal class.

Furthermore, given that federated learning involves the random selection of S nodes for training, there is a probability that the attacker node may not be included in the selected subset, which could hinder the evaluation of the attack's effectiveness. To enable a precise assessment that closely mirrors practical scenarios, the expected value in the Hypergeometric Distribution is utilized. The expected value, or mean, represents the predicted number of successes in a given experiment and is given by $E = (S * A)/P$, where S is the sample size, P is the population size, and A is the number of potential successes (i.e., the number of attacker nodes) within the population. The expected number of selected attacker nodes E in each federated learning round is shown in Table VII. The resulting value is rounded to the nearest integer. During the simulation, E attacker nodes are initially selected at random, and the remaining $(S - E)$ nodes are then randomly chosen from the remaining nodes. Thus, E attack nodes are ensured in each local training round. The attacker behaves normally as a benign node until the global model classification accuracy reaches 0.5. This is to ensure that the global model has already acquired knowledge at a reasonable level from the overall dataset. Thus, the attacker can exploit the global model as the discriminator to generate meaningful fake local data and to train the malicious model subsequently. The malicious model training step is 20 and learning rate is 0.0005.

C. Adaption of Existing Methods

From the defense perspective, trimmed-mean aggregation method is applied to collect and update the global model parameters. There are no extra changes in their settings. The trimmed mean of each parameter is computed by averaging the values of each parameter that has removed the largest and smallest values. For the existing Fang's poisoning attack method, it is adapted to perform the targeted attack and thus the gradient direction is changed to follow the targeted label. The update values are sampled from the intervals of $[\mu + 3\sigma, \mu + 4\sigma]$. Moreover, Fang's method is integrated with the proposed fake data generation process to prepare the poisoned dataset. Overall, both trimmed mean aggregation rule and Fang's attack method have been fine tuned in this paper to ensure their reliability.

D. Metrics

The objective of a targeted poisoning attack is to manipulate the global model such that it classifies the targeted data as a specific, attacker-chosen label. To evaluate the effectiveness of such an attack, two metrics are defined: poison accuracy and main accuracy. The poison accuracy measures the proportion of targeted data that is classified as the attacker-chosen label relative to the total number of targeted data instances. It quantifies the success of the attack in influencing the model to misclassify the targeted instances. The main accuracy represents the proportion of non-targeted data that is correctly classified relative to the total number of non-targeted data instances. It assesses the overall performance of the model on data that was not the focus of the attack. This value highlights the impact of targeted attack to those untargeted classes. In addition, the standard test accuracy is obtained. It highlights the overall global model performance when facing targeted attacks. Furthermore, the survival rate quantifies the proportion of instances in which the attacker's malicious model remains included in the norm-based aggregation, rather than being excluded, relative to the total number of times the attacker node is selected for training. All the metrics are collected after attacker starts the poisoning attack when the global model accuracy reaches 0.5.

V. RESULTS

The evaluation process is done on a GPU server with Nvidia 3060TI of 8GB RAM. The CUDA version is 12.6. Programs

TABLE VIII

SURVIVAL RATE OF ATTACKER NODES AGAINST PROPOSED NORM BASED AGGREGATION RULE

Number of S	**6**			**8**			**10**		
Number of A	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
PoisonGAN	0.01	0.3571	**0.8784**	0.01	0.51	0.7553	0.01	0.51	0.67
Fang's	-	0.3333	0.6031	-	0.5	0.6363	-	0.5	0.6667
TAPA (proposed)	**0.73**	**0.7815**	0.8091	**0.83**	**0.92**	**0.9453**	**0.71**	**0.99**	**0.9833**

TABLE IX

AVERAGE POISON ACCURACY OF 100 ATTACKING ROUNDS AGAINST PROPOSED NORM BASED AGGREGATION RULE

Number of S	**6**			**8**			**10**		
Number of A	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
PoisonGAN	0	0	0.5042	**0**	0	0	**0**	0	0
Fang's	-	0	0.8073	-	0.1247	0.7023	-	0.1533	0.6023
TAPA (proposed)	**0.0051**	**0.7742**	**0.8955**	**0**	**0.7931**	**0.8472**	**0**	**0.5755**	**0.6261**

TABLE X

AVERAGE MAIN ACCURACY OF 100 ATTACKING ROUNDS AGAINST PROPOSED NORM BASED AGGREGATION RULE

Number of S	**6**			**8**			**10**		
Number of A	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
PoisonGAN	0.9474	0.9826	0.6590	0.9691	0.9883	0.8274	0.9890	0.9928	0.8888
Fang's	-	0.9910	**0.9939**	-	0.9931	0.9891	-	0.9953	0.9945
TAPA (proposed)	**0.9870**	**0.9955**	0.9937	**0.9921**	**0.9940**	**0.9946**	**0.9926**	**0.9953**	**0.9954**

TABLE XI

AVERAGE STANDARD ACCURACY OF 100 ATTACKING ROUNDS AGAINST PROPOSED NORM BASED AGGREGATION RULE

Number of S	**6**			**8**			**10**		
Number of A	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
PoisonGAN	**0.9543**	0.9852	**0.6317**	**0.9733**	0.9893	0.8536	**0.9900**	0.9930	0.9061
Fang's	-	0.9915	0.8380	-	0.9411	**0.8470**	-	0.9702	0.8992
TAPA (proposed)	0.9915	**0.8699**	0.8472	0.9927	**0.8649**	0.8481	0.9930	**0.9015**	0.**8933**

TABLE XII

SURVIVAL RATE OF ATTACKER NODES AGAINST TRIMMED MEAN AGGREGATION RULE

Number of S	**6**		**8**		**10**	
Number of A	**2**	**3**	**2**	**3**	**2**	**3**
PoisonGAN	0.98	**1**	**1**	**1**	**1**	**1**
Fang's	**1**	**1**	**1**	**1**	**1**	**1**
TAPA(proposed)	**1**	**1**	**1**	**1**	**1**	**1**

TABLE XIII

AVERAGE POISON ACCURACY OF 100 ATTACKING ROUNDS AGAINST TRIMMED MEAN AGGREGATION RULE

Number of S	**6**		**8**		**10**	
Number of A	**2**	**3**	**2**	**3**	**2**	**3**
PoisonGAN	0	0.0747	0	0	0	0
Fang's	0	**0.8792**	**0.3388**	**0.732**	0	**0.461**
TAPA(proposed)	**0.199**	0.5571	0.0894	0.2038	0	0

and scripts are using TensorFlow version 2.16.0. From the results, the usual benign local training takes about 30s for each training round. When conducting attack uses TAPA method, the training time takes about 120s for each attacker node in each local training round. When using PoisonGAN method, the training time is 40s for attacker node. This is because PoisonGAN only generates targeted class data. Thus, it takes shorter time than TAPA method. As for Fang's method, it takes an average of about 200s for each attacker node in each local training round. This is because Fang's method requires to check through all the attacker models' parameters to compute the mean and standard deviations. Thus, this process takes a longer time.

While initial experiments of TAPA and PoisonGAN include an extreme case with only one attacker, our findings indicate that successful targeted attacks generally require at least two compromised clients, as shown in Tables VIII to IX. Those results are collected when using the proposed norm based aggregation rule. On the other hand, since Fang's method relies on the computation of the mean and standard deviation of model parameters across models, it is more meaningful to evaluate it in settings with at least two compromised models. Therefore, experiments involving Fang's attack and the trimmed mean aggregation rule are conducted under scenarios with two and three attacker clients. This setup provides a more realistic context and enables fairer and more meaningful comparisons with the proposed TAPA and PoisonGAN methods. Thus Fang's method is not tested for the setting of $A = 1$. The corresponding entries are intentionally marked as "–" in Tables VIII to X. Similarly, Tables XII to XV do not include the settings of A=1. Those results are collected when using the trimmed mean aggregation rule.

A. Norm-Based Aggregation Rule

The implementation of the proposed norm-based aggregation rule ensures that the training of the standard classification

TABLE XIV

AVERAGE MAIN ACCURACY OF 100 ATTACKING ROUNDS AGAINST TRIMMED MEAN AGGREGATION RULE

Number of S	**6**		**8**		**10**	
Number of A	**2**	**3**	**2**	**3**	**2**	**3**
PoisonGAN	0.9823	0.8013	0.9893	0.9229	0.9906	0.9418
Fang's	0.9957	0.9967	0.9957	0.9966	0.9956	0.9978
TAPA(proposed)	**1**	**1**	**1**	**1**	**1**	**1**

TABLE XV

STANDARD MODEL ACCURACY OF 100 ATTACKING ROUNDS AGAINST TRIMMED MEAN AGGREGATION RULE

Number of S	**6**		**8**		**10**	
Number of A	**2**	**3**	**2**	**3**	**2**	**3**
PoisonGAN	0.9783	**0.7720**	0.9850	0.9194	**0.9878**	0.9403
Fang's	0.9955	0.8413	**0.9395**	**0.8773**	0.9941	**0.9192**
TAPA(proposed)	**0.9676**	0.9101	1	0.9668	1	1

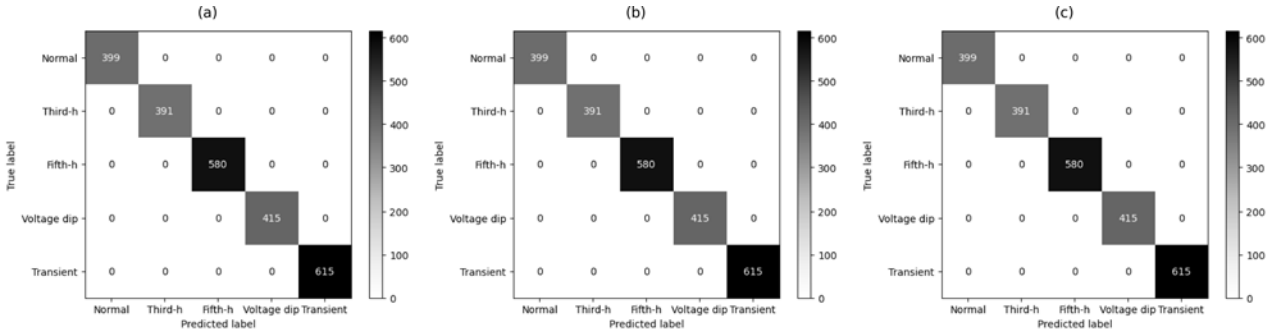

Fig. 3. Confusion matrix with norm-based aggregation rule with no attack, under S/P = (a) 6/10, (b) 8/10, (c) 10/10 federated learning settings.

model in the federated learning process remains unaffected. The confusion matrix of the trained global model on testing data is shown in Fig. 3. The model achieves an accuracy of 100% for all S/P federated learning settings, demonstrating that the norm-based aggregation rule effectively handles various cases without compromising standard performance. As for the mitigating effects on the typical PoisonGAN attack, the poison accuracy is shown in Table IX. Poison accuracy is recorded at each round after the poisoning attack starts. The attacks continues for 100 rounds and the average poison accuracy is thus obtained. It can be seen that PoisonGAN fails to influence the global model, and all poison accuracies are 0 except for the case of $A/S = 3/6$ federated learning setting. The poison accuracy of 0.5042 under $A/S = 3/6$ federated learning setting reveals that this attack method requires attackers to compromise a majority of the local nodes to increase the likelihood of success. This shows that the norm-based aggregation rule is effective to identify and remove the impacts of the poisoning models that have magnified malicious model parameters.

B. Survival Rate

The results of the survival rate are summarized in Table VIII and Table XII for the proposed norm-based and existing trimmed mean method respectively. The best values of each setting are made bold. Higher survival rate indicates more malicious models have bypassed the defense and have been used in the aggregation process. It is noted that the proposed norm-based aggregation rule has reduced the survival rate from PoisonGAN method. The survival rates are approximately 0.5 or below when the number of attacker nodes is 1 and 2. Only when there are 3 attacker nodes, the malicious model parameters from PoisonGAN have a higher chance to break the robust aggregation rule. When there are 3 attacker nodes with selection number $S = 6$, the survival rate is as high as 0.8784, which is better than the proposed TAPA method. This particular result is explained in Section V-C. Nevertheless, these results prove the effectiveness of the proposed norm-based aggregation rule to remove malicious model parameters with relatively large or small norms.

For the proposed TAPA method, it is evident that the malicious models exhibit substantially higher survival rates, consistently exceeding 0.7. Notably, even with a single attacker node and a selection number $S = 6$, the survival rate reaches as high as 0.73. The highest observed survival rate is 0.99, occurring with 2 attacker nodes and a selection number $S = 10$. This value indicates that the attacker nodes successfully bypassed the aggregation defense mechanism at nearly every training round. These results demonstrate that the proposed malicious training method significantly enhances the survival rate of malicious parameters within a robust aggregation rule.

Survival rates when attacking attacking trimmed mean method are in Table XII. It can be seen that for trimmed mean method, all attacker nodes can achieve a survival rate of 1 in almost all situations. This shows that trimmed mean method does not repel the malicious model as whole, and it will still use the malicious models' parameters if they are not the largest or smallest among all models. The proposed

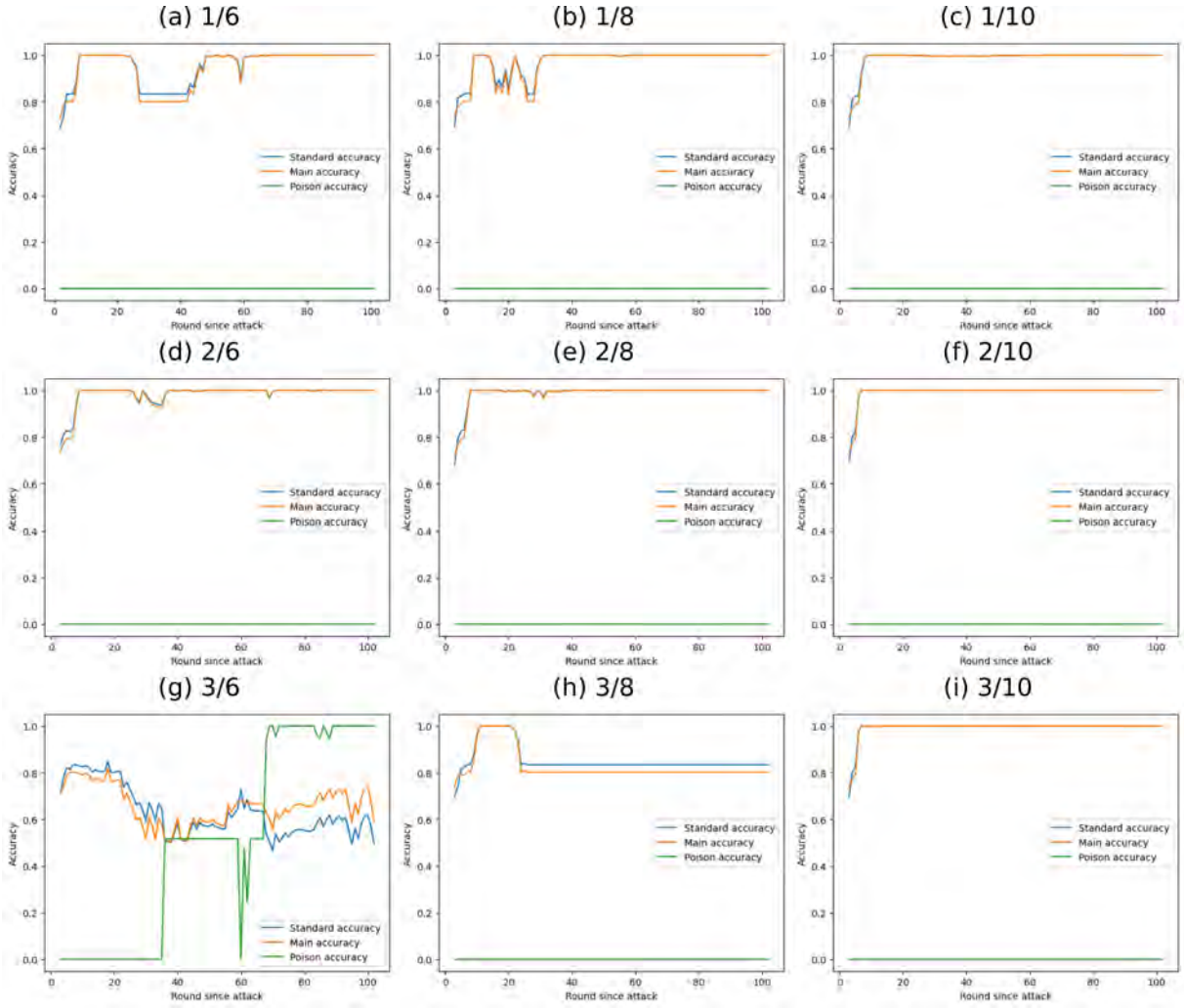

Fig. 4. PoisonGAN: Attack performance under different A/S settings over federated learning rounds.

norm based method rejects the whole model if the model's norm is extremely large or small. This is evident from the smaller values of survival rate in Table VIII. The survival rates have clearer distinctions between attack methods. While TAPA consistently maintains a large survival rate across all scenarios, Fang's method struggles with fewer attackers. For example, when $S = 6$ and $A = 2$, Fang's method only survives in 33% of rounds.

C. Poisoning Attack Performance

1) Poisoning Attack Trends: The attack performances over 100 attacking rounds are illustrated in Fig 4 (PoisonGAN), 5 (proposed TAPA) and 6 (Fang's). The blue line is the overall model's standard accuracy. The orange line the main accuracy. The green line is the poison accuracy. Due to space constraint, Fig 6 only shows the trends for Fang's method under the settings of $A = 3$ and $S = 6$. Nevertheless it presents the Fang's best attack performance among its experiments and helps to provide the performance insight of this method. The results in Fig. 4 indicate that the PoisonGAN method is ineffective in performing targeted poisoning attacks across most settings. Because the poison accuracy remains low and main accuracy and standard accuracy remain at high values for most of the case. This shows the unsuccessful targeted poisoning attack and no targeted data has been misclassified. PoisonGAN succeeds in targeted attack only when $A = 3$ and $S = 6$. In such setting, the expected number of attacker nodes is 2, coinciding with a lower number of benign local nodes, which allows the poison accuracy to gradually increase as shown in Fig. 4(g). However, this increment in poison accuracy is accompanied by a decline in main accuracy, demonstrating that PoisonGAN does not effectively maintain high main accuracy as intended. It has impacted the non-targeted classes as well and thus made the attack to be outstanding to central server. It is observed that the poison accuracy initially rises to approximately 0.5 and maintains this level for about 30 attacking rounds. Subsequently, it increases to 1 after 60 rounds from the start of the attack. In contrast, the main accuracy declines from 0.8 to 0.5 and then stabilizes around 0.6. This trend indicates that as poison accuracy increases and main accuracy decreases, the global model is increasingly biased towards classifying all input data as the attacker-chosen label. Consequently, the global model

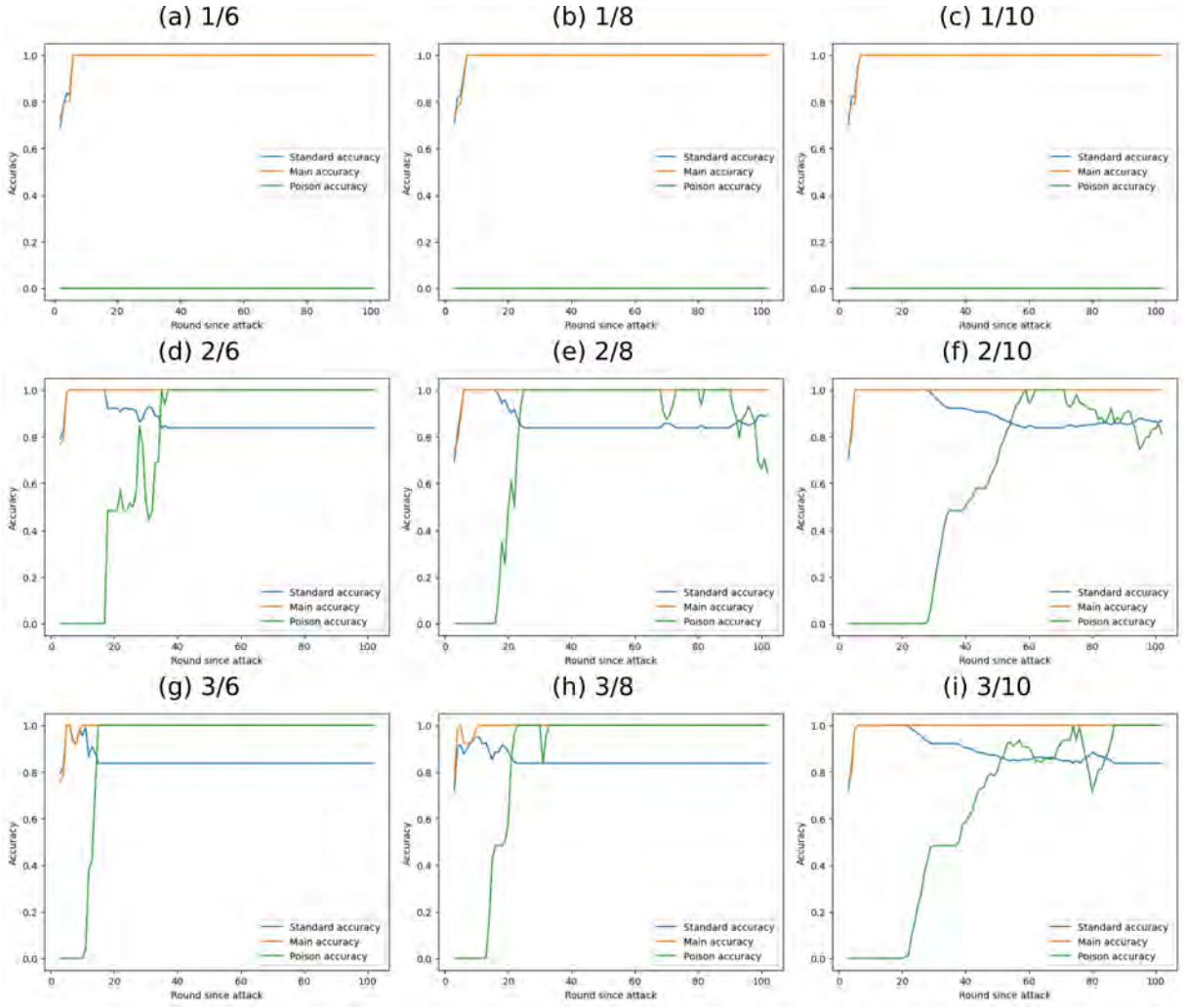

Fig. 5. TAPA: Attack performance under different A/S settings over federated learning rounds.

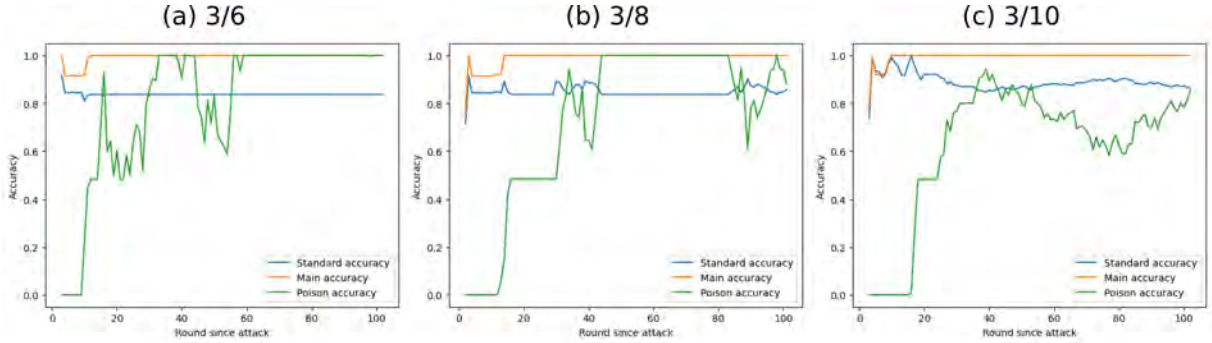

Fig. 6. Fang's: Attack performance under different A/S settings over federated learning rounds.

parameters become similar to the malicious model parameters over the federated learning rounds, enhancing the malicious model's likelihood of surviving in the norm-based aggregation rule. This observation explains the high survival rate of 0.8784 reported in the previous section. The confusion matrix, as shown in Figure 7(a), further supports this analysis, revealing that the global model fails to learn the patterns of the targeted data and predominantly classifies test data as the attacker-chosen label. This indicates the failure of PoisonGAN to perform the targeted poisoning attacks that does not affect the main accuracy. Further analysis of Fig. 4(a) to Fig. 4(f) reveals that when the number of attacker nodes is small, poison accuracy remains at 0, while main accuracy steadily approaches and remains at 1. In contrast, Fig. 4(h) and Fig. 4(i) show a notable drop in main accuracy, even though poison accuracy remains at 0. This decline in main accuracy is also due to the training of the malicious model using data from a single class, which prevents the model from learning crucial patterns from the poisoned data, thereby negatively impacting the overall accuracy. The plots also show the correlation

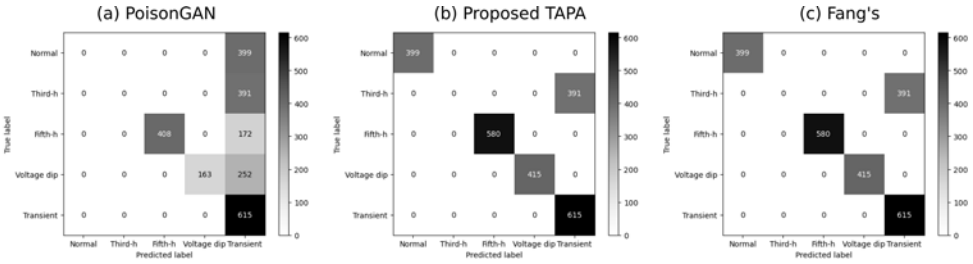

Fig. 7. Confusion matrix from global model after poisoning attack using (a) PoisonGAN and (b) TAPA method (C) Fang's method, in federated learning setting of A/S = 3/6.

between main accuracy and standard accuracy. Their trends are almost the same. Because the main accuracy represents the accuracy of all untargeted classes and contributes the major parts into standard accuracy computation. Thus, they follow the same trend.

The results presented in Fig. 5 demonstrate that the proposed TAPA method performs much better than PoisonGAN. Under all selection scenarios, the poison accuracy reaches 1 when there are two or more attacking nodes. These show that TAPA has successfully poisoned global model to misclassify targeted class to a chosen new class. Notably, the main accuracies across all nine settings reach 1 and consistently maintain this level throughout the federated learning process. This indicates that the TAPA method effectively executes poisoning attacks without compromising the main accuracies of the global model. All the other non-targeted classes are classified correctly by the global model. The plots also show that the standard accuracy drops when poison accuracy increases. This is inevitable, because this attack has led global model to misclassify targeted class, the overall model standard accuracy drops. Nevertheless, the global model is only poisoned to affect the targeted data, the model's behavior on non-target classes largely remains unaffected. This demonstrates that the attack is precise in its objective, despite introducing some collateral accuracy loss. The trends in poison accuracies observed in Fig. 5(d) to Fig. 5(i) correspond with the high survival rates previously discussed. Conversely, Fig. 5(a) to Fig. 5(c) show low poison accuracies of 0 due to the presence of only a single attacker node. Despite the high probability of this single malicious model surviving the aggregation rule, its impact is diminished when combined with the parameters of benign local models. Furthermore, the decline in poison accuracies in Fig. 5(e) and Fig. 5(f) results from a reduced number of surviving malicious models, with their effects being mitigated by the significantly larger number of benign models. Moreover, it can be seen that the proposed TAPA method can have poison impacts quicker at a higher A/S ratio. For example, the round that have non-zero poison accuracy is at round 20 when $A = 3$ and $S = 6$. But the round that have non-zero poison accuracy is at round 35 when $A = 3$ and $S = 10$. Additionally, Fig. 7(b) shows the classification results from the global model after poisoning attack. It can be seen that the global classification model has been totally affected by the malicious model. The targeted data of "Third-h" (label 1) are entirely misclassified as "Transient" (label 4), while all other data are correctly classified according to their true labels.

Fang's method's performance trend is shown in Fig. 6. Fang's method gets high poison accuracy across all settings. The poison accuracy can reach to 1 and the main accuracy maintain at around 1 as well. However, there are more fluctuations in the poison accuracy over the attacking rounds, as compared to the results of TAPA's attack performance, under the same setting of $A = 3$ and $S = 6$. Especially for the setting of $A = 3$ and $S = 10$, the poison accuracy reaches 0.9 at round 40 and then gradually decreases. At round 80, the poison accuracy increases again. These results shows that although the Fang's method is effective to perform targeted poisoning attack, its performance is not stable over attacking rounds. However, we can see the proposed TAPA method can have more stable performance over attacking rounds. Moreover, Fig. 7(c) shows the classification results from the global model after Fang's attack is performed under the setting of $A = 3$ and $S = 6$. It can be seen that the global classification model has been totally affected by the malicious model. The targeted data of "Third-h" (label 1) are entirely misclassified as "Transient" (label 4), while all other data are correctly classified according to their true labels. This corresponds to the poison accuracy trend as shown in Fig. 6 (a). The global model is finally poisoned with poison accuracy of 1. Similarly, the standard accuracy and the main accuracy of the global model under Fang's attack have the same trend as the results in proposed TAPA method. Main accuracy remains at high and standard accuracy decreases when poison accuracy increases. Those indicates the targeted attack is successfully conducted, when using TAPA and Fang's method.

In conclusion, the TAPA method successfully performs targeted poisoning attacks with both high poison accuracy and main accuracy, demonstrating its effectiveness in maintaining attack precision while preserving overall model performance. The Fang's method can achieve the similar results as well, but with less stable performance.

2) Average Poison Accuracy: To better quantify and evaluate the attack performance results, the average poison accuracy, main accuracy and standard accuracy are computed across the attacking rounds. Table IX to XI summarize the results when using proposed norm based aggregation rule and Table XIII to XV summarize the results when using trimmed mean based aggregation rule. The best value of each column is made bold.

As shown in Table IX, TAPA achieves high poisoning accuracy even with only two attackers. For instance, when $S = 6$ and $A = 2$, TAPA reaches 0.7742 poisoning accuracy,

whereas both Fang's method and PoisonGAN fail entirely. This performance advantage appears across all tested values of S and A. As the number of attackers increases from 2 to 3, TAPA's poisoning accuracy remains stable, further showcasing its efficiency. Under the trimmed mean rule as shown in Table XIII, poisoning success drops for the proposed TAPA method. The highest poisoning accuracy it can achieve is 0.5571 when $S = 6$ and $A = 3$. It totally fails when the number of selected model S is 10. Nevertheless, the proposed TAPA method is the only one to have poisoning accuracy of 0.199 even with $S = 6$ and $A = 2$. Other two methods both fail under such adverse scenarios. It shows that the proposed method is reliable when there is less compromised attacker nodes. It is also noted that Fang's method performs better against the trimmed mean aggregation rule, as shown in Table XIII. It has the highest poisoning accuracy of 0.8792 when when $S = 6$ and $A = 3$. Fang's method has better performance than the proposed TAPA method among most of the settings in trimmed mean aggregation rule. It benefits from the design of escape coordinate-wise trimming, specifically for trimmed mean method. However, it has the disadvantages that it may perform worse when there are less attacker nodes. By comparing Table IX and Table XIII, it can be seen that Fang's method has higher poison accuracy against trimmed mean aggregation rule. When it faces the proposed norm based aggregation rule, the poison accuracy decreases. For example the value drops from 0.8792 to 0.8073 when $S = 6$ and $A = 3$. This demonstrates that the proposed aggregation rule can reduce the impacts of the Fang's method in performing the targeted attack in this paper. As for the PoisonGAN method, it remains mostly ineffective, achieving near-zero poisoning success in all cases. Overall, it could be seen that there is no attack method that performs well in all settings. And in general, it can be concluded that the higher ratio of A/S, the better poison accuracy of the attacks.

3) Average Main Accuracy: The main accuracies under the proposed norm based rule are shown in Table X. The highest value of each column is made bold. TAPA maintains high main task accuracy. It has main accuracies exceeding 0.99 across all settings. This indicates that the infected global model has non-degraded performance on legitimate inputs. The TAPA method preserves its utility during attack. Fang's method also maintains high main task accuracy, though slightly lower than TAPA in some configurations. For example, with $S = 8$ and $A = 2$, Fang's method achieves 0.9931, compared to 0.9940 for TAPA. PoisonGAN, in contrast, consistently causes more harm to the main task, with main accuracies dropping to around 0.8274 when $S = 8$ and $A = 3$. When under trimmed mean aggregation rule as shown in Table XIV, the proposed TAPA method has 100% main accuracy in all situations. This also attributed to the design of targeted poisoned data, and to train a model that will not affect the untargeted classes. Although TAPA does not guarantee high targeted poisoning accuracy in all settings, this result demonstrates the effectives of the proposed attack process. It also can be seen that the Fang's method maintains stable performance to achieve main accuracy exceeding 0.99 for all settings. This result shows that Fang's method is reliable against the trimmed mean aggregation rule. PoisonGAN method has lower main accuracy ranging from 0.8013to 0.9906. This indicates that while PoisonGAN is still able to perform attacks, it lacks precision and tends to degrade the model's utility by affecting benign inputs.

4) Average Standard Accuracy: Standard accuracy refers to the overall prediction accuracy evaluated on the full test dataset, which includes both clean (benign) and poisoned (malicious) samples. In terms of a successful poisoning attack regardless of targeted or untargeted attack, the value of this standard accuracy should be as low as possible. In Table XI, under the proposed norm-based aggregation rule, the TAPA method consistently achieves lower final accuracy of the standard model compared to PoisonGAN and Fang's method. The lowest value of each column is made bold.The standard accuracy ranges from about 0.8472 to 0.9015. This suggests that TAPA is more effective at degrading model performance, indicating stronger attack potency. Fang's method can also introduce the lowest accuracy of 0.847 when $S = 8$ and $A = 3$. It is also observed that, although the PoisonGAN method does not successfully achieve its targeted attack objective, it still disrupts the model by inducing misclassifications in an untargeted manner. For example under the setting of $A = 3$ and $S = 8$, the poison accuracy is 0 but the main accuracy is 0.8274 and standard accuracy is 0.9194. This method degrades the global model to have the lowest standard accuracy of 0.6317 across all settings as well. Table XV shows the results under trimmed mean aggregation rule. Since TAPA has relatively worse attack performance against this aggregation rule, the standard accuracies are higher. The values range from 0.9101 to 1. It aligns well with its poison accuracies. As for Fang's method, its attack performance is generally better than the other two methods and as shown in Table XIII, it has the lowest standard accuracies across most settings. It has a value of 0.8413 when $S = 6$ and $A = 3$. It is noted that although PoisonGAN method does not have high poison accuracies, it can reduce the standard accuracy effectively. When $S = 6$ and $A = 3$, the value is 0.772. This result shows that PoisonGAN has an impact on the global model under trimmed mean aggregation rule. It is able to poison the global model to misclassify data, but in an unexpected and untargeted way.

All the results demonstrate that both our proposed TAPA attack and Fang's method exhibit strong performance of targeted poisoning attacks, but in complementary ways. Fang's method is particularly effective under the trimmed mean aggregation rule, as it is specifically designed to evade parameter-wise trimming. However, it suffers under the proposed norm based aggregation rule with reduced attack success rate, especially when the number of compromised clients is small, due to its reliance on statistical assumptions that break down with limited data. The poison accuracies decrease under such settings. In contrast, TAPA shows robust performance in aggregation strategy of model-wise trimming. It achieves high poisoning success while maintaining model utility, even with a small number of attackers. TAPA does not rely on statistical assumptions about the distribution of benign or malicious updates, making it more resilient to aggregation defenses and effective even with a small number of attackers.

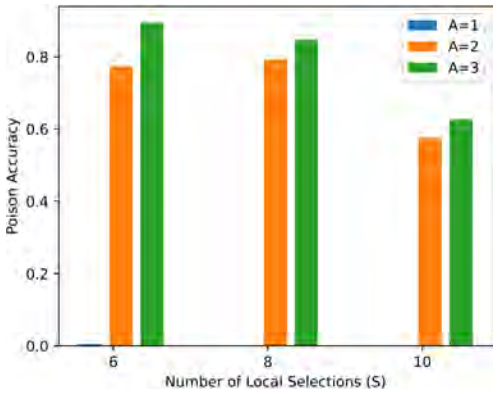

Fig. 8. Poison accuracy for different attacking node number A when local selection number S is fixed.

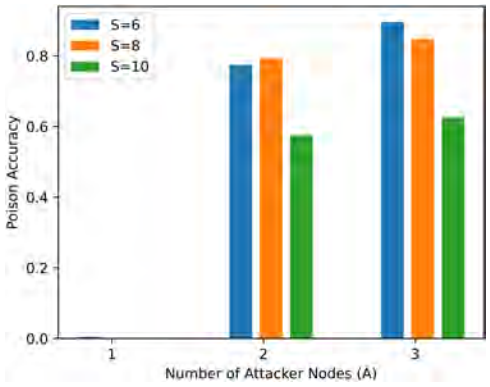

Fig. 9. Poison accuracy for different local selection number S when attacking node number A is fixed.

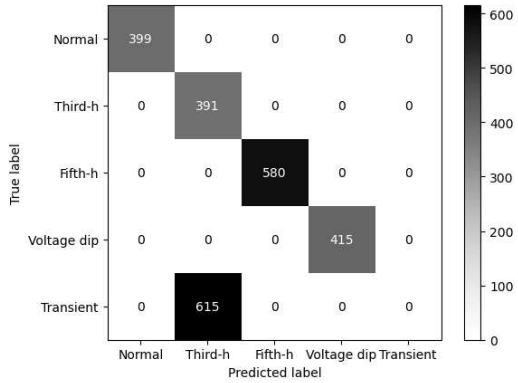

Fig. 10. Confusion matrix when flipping "Transient" (label 4) to "Third-h" (label 1) using proposed method under federated learning setting of A/S=3/6.

TABLE XVI
PROPORTION OF TARGETED CLASS DATA IN EACH LOCAL DATASET

Node	Class 1	Class 4
0	0.0978	0.1213
1	0.1762	0.2070
2	0.1446	0.3052
3	0.1000	0.5550
4	0.1765	0.0000
5	0.0000	0.3840
6	0.0000	0.3704
7	0.1982	0.0901
8	0.6612	0.0000
9	0.1951	0.5122

This makes TAPA more adaptable in diverse federated learning environments. PoisonGAN, while capable of achieving notable poisoning success in certain configurations, generally exhibits less precision. It has the cost of degrading the main task accuracy and increasing detection risk.

5) Number of Attacker Nodes: To evaluate the impact of the number of attacker nodes (A) and the local selection number (S) when using the TAPA method to attack against proposed norm based aggregation rule. Specific analysis is done as shown in Fig. 8 and Fig. 9. Figure 8 provides a clear understanding of different A when S is fixed. It is observed that for a fixed S, increasing the number of attacker nodes A leads to higher poison accuracy. This is expected, as a greater number of attacking models amplifies the influence of malicious parameters and enhances their survival rate in the aggregation process. Additionally, it is important to note that having only a single attacking node does not significantly affect the poisoning attacks. Figure 9 illustrates that when A is fixed, increasing S reduces the poison accuracy. This is intuitive, as the presence of more benign local models dilutes the influence of the malicious models. Nevertheless, the proposed TAPA method can poison the global model successfully using small number of attacker nodes, and the minimum requirement is to compromise just 2 local nodes.

6) Quantity of Targeted Data: While Fig. 7 (b) represents the successful results when flipping class 1 "Third-h" to class 4 "Transient" in test data, it is the result of changing label from a class with a smaller number of data point to a class with a larger number of data points. Conversely, Fig. 10 shows the results when the attack is reversed, with class "Transient" being changed to class "Third-h". It is shown that the entire "Transient" data in test set has been classified as "Third-h" data, from the more frequent class to those of the less frequent class. Thus, these findings confirm the robustness of the proposed method, demonstrating its capability to address poisoning attacks across various data distributions, regardless of data size. The specific ratios of targeted data class 1 and class 4 in each local group are summarized in Table XVI. The distributions of class 1 and class 4 across different clients are highly non-uniform. Specifically here the attacker tries to use class 1 data as targeted class to change it to the other class. For node 5 that is compromized by attacker, there is no targeted data. Attacker can generate the non-existing targeted class 1 data in node 5 and train the malcious model using the proposed TAPA method. Importantly, the proposed method maintains consistent performance regardless of the direction or severity of the non-IID data skew. Whether the attack involves flipping labels from a minority class to a majority class, or vice versa, the method demonstrates strong resilience.

D. Future Directions

While our proposed TAPA method demonstrates the potential risks of targeted poisoning in power quality classification systems, it also highlights the urgent need for more robust defense mechanisms. Future research could explore adaptive aggregation techniques beyond the norm-based and trimmed mean rules. One straightforward method is to have the validation dataset at central server. This dataset can be used to

examine the performance of each collected local model and to identify any malicious local node and to exclude that node permanently. The proposed fake data generation process in this paper may also be used as the countermeasure. Whether the central server can obtain the real validation data or not, they can use the generation process to get synthetic fake data. Such data will follow the hidden patterns of all data and provide sufficient insights of the local model's behavior. However, it is important to note that this approach relies on the integrity of the global model. If the global model has already been compromised or poisoned at an early stage, the generated validation data may reflect biased or misleading patterns, thus diminishing the effectiveness of this countermeasure. On the other hand, federated learning frameworks may also benefit from integrating behavioral monitoring of model updates over time to flag sudden or suspicious changes. To mitigate this risk, future systems might incorporate ensemble models or delayed aggregation, where updates are tested across multiple versions of the global model before final inclusion. Meanwhile, secure bootstrapping techniques can also be used. The server may initialize the global model using a small, trusted dataset or pre-train it on a generic but relevant public dataset. This gives the model an initial direction and reduces vulnerability to extreme updates. Another strategy is the consistency monitoring of updates from clients. They are either simulated, sandboxed, or subjected to stricter scrutiny before being incorporated into global model. Instead of aggregating blindly, the server could delay full update integration until it observes a consensus trend across multiple independent clients. By implementation of the suggested strategies and any other methods, the resilience of the federated learning framework against poisoning attacks can be enhanced.

VI. Conclusion

While federated learning systems address data privacy and security issues in smart grids, they remain vulnerable to poisoning attacks aimed at degrading model performance. To improve robustness, we initially introduced a norm-based aggregation rule that excludes model parameters with extreme norms, either excessively large or small, updating the global model solely based on the remaining local model parameters. Experimental results demonstrate that this straightforward rule effectively mitigates the impact of poisoning attacks, especially those that amplify malicious parameters to impose greater influence on the global model. Additionally, from the perspective of the attacker, we proposed TAPA which leverages generated fake samples to train a malicious model. Unlike existing approaches, TAPA generates fake data from all classes and integrates them with the original local data, with targeted data labels altered to any attacker-chosen label. The malicious model is then trained on this augmented poisoned dataset, with a focus on aligning its norm closely with the global model to increase the likelihood of selection by robust aggregation rules during the update process. Evaluation results show that the proposed poisoning attack strategy achieves a higher survival rate under secure aggregation rules, thereby successfully executing targeted attacks. Future work will involve validating the proposed poisoning attacks against other robust aggregation rules to assess the generalizability of this approach, and a more robust aggregation rule to counter the proposed attack method.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, 2017, pp. 1273–1282.

[2] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: A federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.

[3] M. M. Badr et al., "Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7719–7736, May 2023.

[4] H. Liang, D. Liu, X. Zeng, and C. Ye, "An intrusion detection method for advanced metering infrastructure system based on federated learning," *J. Modern Power Syst. Clean Energy*, vol. 11, no. 3, pp. 927–937, 2023.

[5] N. Waleed, I. Emad, M. Anany, and W. Rady, "Integration of federated machine learning in smart metering systems," in *Proc. 4th Novel Intell. Lead. Emerg. Sci. Conf. (NILES)*, Oct. 2022, pp. 35–38.

[6] H. N. C. Neto, J. Hribar, I. Dusparic, D. M. F. Mattos, and N. C. Fernandes, "A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends," *IEEE Access*, vol. 11, pp. 41928–41953, 2023.

[7] E. Hallaji, R. Razavi-Far, M. Saif, B. Wang, and Q. Yang, "Decentralized federated learning: A survey on security and privacy," *IEEE Trans. Big Data*, vol. 10, no. 2, pp. 194–213, Apr. 2024.

[8] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.

[9] Y. Zhu, H. Wen, R. Zhao, Y. Jiang, Q. Liu, and P. Zhang, "Research on data poisoning attack against smart grid cyber–physical system based on edge computing," *Sensors*, vol. 23, no. 9, p. 4509, May 2023.

[10] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, and F. Piccialli, "Model aggregation techniques in federated learning: A comprehensive survey," *Future Gener. Comput. Syst.*, vol. 150, pp. 272–293, Jan. 2024.

[11] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 7, pp. 8726–8746, Jul. 2024.

[12] H. Xiao, X. Huang, and C. Eckert, "Adversarial label flips attack on support vector machines," in *Proc. ECAI*, 2012, pp. 870–875.

[13] X. Cao and N. Z. Gong, "MPAF: Model poisoning attacks to federated learning based on fake clients," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3395–3403.

[14] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2020, pp. 480–501.

[15] D. Rong, S. Ye, R. Zhao, H. N. Yuen, J. Chen, and Q. He, "FedRecAttack: Model poisoning attack to federated recommendation," in *Proc. IEEE 38th Int. Conf. Data Eng. (ICDE)*, May 2022, pp. 2643–2655.

[16] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3310–3322, Mar. 2021.

[17] Y. Zhang, Y. Zhang, Z. Zhang, H. Bai, T. Zhong, and M. Song, "Evaluation of data poisoning attacks on federated learning-based network intrusion detection system," in *Proc. IEEE 24th Int. Conf. High Perform. Comput. Commun., 8th Int. Conf. Data Sci. Syst., 20th Int. Conf. Smart City, 8th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Dec. 2022, pp. 2235–2242.

[18] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1605–1622.

[19] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. NeurIPS*, vol. 30, 2017, pp. 1–16.

[20] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 5650–5659.

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

[22] A. V. Vidyapeetham and Honey well, "Power quality classification dataset—1," Amrita Honeywell Hackathon 2021 Dataset, Bengaluru, India, 2021. [Online]. Available: https://www.kaggle.com/datasets/jaideepreddykotla/powerqualitydistributiondataset1