FedCWA: Credibility-Weighted Aggregation for Byzantine-Robust Federated Learning

Ibrahim Musa[1], Shaoxi Teng[1], Xutong Mu[1]
[1]School of Computer Science and Technology, Xidian University, Xi'an, 710071, China

Federated learning (FL) is susceptible to Byzantine attacks, where malicious clients can corrupt local data or upload adversarial updates to undermine model training. Many existing defense methods assume that data often rely on data homogeneity assumptions or prior datasets. To overcome these issues, we present FedCWA, a credibility-weighted aggregate framework for Byzantine-robust federated learning based on credibility-weighted aggregation. FedCWA presents ProfDiff, a technique that generates a fair proxy dataset (PDFD) on the server based on client class prototypes, which represents the global data distribution, eliminating dependency on external prior datasets. By analyzing the similarity of client prediction behaviors on PDFD, we construct a logits similarity matrix based on cosine similarity, enabling fine-grained client credibility assessment. Depending on the assessment results, the scheme designs a dynamic weight optimization mechanism that adaptively adjusts aggregation weights to effectively suppress the influence of malicious clients. Comprehensive experiments achieved on different benchmark datasets under a variety of Byzantine attack scenarios demonstrate that FedCWA consistently outperforms existing state-of-the-art defense methods, achieving higher accuracy, improved stability in convergence, and greater resilience in heterogeneous federated learning settings. Theoretical analysis further substantiates the robustness guarantees of our methodology, achieving FedCWA as an efficacious strategy for protecting federated learning.

***Index Terms*—Federated learning, Byzantine attacks, Credibility-weighted aggregation, Malicious clients.**

I. Introduction

FEDERATED Learning (FL) [1] has emerged as a novel distributed machine learning framework that enables multiple clients to collaboratively train a global model without exposing raw local data. By transmitting model updates instead of raw data samples, FL mitigates privacy risks while leveraging distributed edge resources. This decentralized framework has attracted growing interest in several fields, such as banking [2], healthcare [3], and autonomous systems [4], due to its ability to preserve data confidentiality, reduce communication overhead, and scale across heterogeneous devices and data sources.

Despite these advantages, FL remains highly susceptible to Byzantine attacks, in which malicious clients deliberately submit corrupted information to undermine global training. These adversaries may manipulate local datasets (e.g., through label flipping or poisoning [5]), insert adversarial perturbations into model updates (e.g., sign-flipping scaling [6] or backdoor triggers [7]), or exploit collusion to enhance their impact. These attacks can severely degrade global model accuracy, destabilize convergence, or even provoke harmful decision behaviors.

To counter these attacks, researchers have proposed multiple Byzantine-robust aggregation techniques [8],[9],[10],[11]. Existing Byzantine defense strategies can be categorized into three primary types: 1) Distance-based techniques such as Krum [12], Multi-Krum [13] and Foolsgold [14]. These techniques detect abnormal updates by measuring distances across client models; however, they can misclassify benign clients with special data distributions in heterogeneous environments. 2) Statistical distribution-based techniques such as Trimmed Mean [15], Bulyan [16], and RFA [17]. These techniques are used to eliminate anomalous updates based on statistical characteristics; however, they often require strong assumptions regarding data distributions and a significantly greater quantity of benign clients compared to malicious ones. 3) Prior dataset-based methods such as FLTrust [18], Sageflow [19], and SDEA [20]. These techniques, while effective, suffer significant challenges in obtaining costly, high-quality manually annotated proxy datasets that accurately reflect the real application area.

In summary, whereas existing defenses offer limited robustness, they encounter tremendous limitations in heterogeneous federated learning environments. Distance-based approaches suffer under non-IID distributions, statistical -based approaches rely on strict assumptions, and prior-based dataset methods face challenges regarding practicality and data acquisition. These limitations degrade both universality and robustness, motivating the development of a more reliable and statistically grounded defense.

This paper proposes a Credibility-Weighted Aggregation (CWA) based Byzantine-robust Federated Learning approach, termed FedCWA. The method offers two key advantages: (i) it dynamically constructs a fair proxy dataset by leveraging class prototype information uploaded by clients, thereby removing reliance on external prior datasets; this "fairness" refers to ensuring balanced coverage of all classes and avoiding bias toward any single client's data distribution; and (ii) it enables fine-grained credibility assessment by analyzing client prediction behavior on this proxy dataset, which strengthens defense effectiveness in heterogeneous environments. To provide a more intuitive understanding, client credibility measures the consistency of each client's predictions on the proxy dataset relative to the aggregated benign behavior. Clients whose predictions align well with the majority trend are assigned higher credibility, indicating more reliable behavior. Unlike traditional

Manuscript received June 18, 2025; revised August 29, 2025. Corresponding author: Xutong Mu (email: muxutong@xidian.edu.cn).

similarity measures that focus solely on update distances or gradient norms, credibility directly reflects prediction-level behavioral alignment, making it more robust under model heterogeneity. Technically, FedCWA first uses client-provided class prototypes to generate a proxy dataset (PDFD), then computes credibility weights by evaluating prediction consistency on this dataset, and finally applies a dynamic weight-optimization mechanism to ensure fair contribution from benign clients.

In summary, this paper makes the following contributions

- We introduce FedCWA, a credibility-weighted aggregation–based Byzantine-robust federated learning framework. It incorporates ProDiff, a class-prototype–driven method that dynamically generates a fair proxy dataset (PDFD) on the server, thereby eliminating reliance on external prior datasets.
- FedCWA analyzes client prediction behavior on the PDFD, constructs a logits-similarity matrix via cosine similarity, and derives fine-grained credibility scores to guide aggregation—effectively suppressing the influence of malicious clients under heterogeneous settings.
- We conduct extensive experiments on multiple benchmark datasets across diverse Byzantine attack scenarios. The results show that FedCWA outperforms state-of-the-art defenses, yielding notable improvements in accuracy and convergence stability in heterogeneous federated learning environments.

Organization. The rest of this paper is organized as follows. Section II reviews Byzantine attacks and defense mechanisms in federated learning. Section III presents the motivation and details of the proposed FedCWA framework. Section IV describes the experimental setup, and Section V provides an in-depth analysis of the experimental results. Finally, Section VI concludes the paper.

II. Related works

This section provides a brief description of Byzantine attacks in federated learning, followed by a review of current research on Byzantine-robust federated learning approaches.

A. Byzantine Attacks to FL

In federated learning, Byzantine attacks occur when malicious clients interfere with local data or injecting malicious parameters to the training process. The result may render the model inaccurate, hamper convergence, or even stop it from convergence completely. Byzantine attacks can be classified into two main categories based on how they work: data-based attacks and parameter-based attacks.

1) Data-based Attacks

Huang et al. [21] presented the label flipping attack, in which malicious clients systematically swap accurate labels in their local datasets with inaccurate ones, leading the global model to establish incorrect decision boundaries. This attack particularly harmful in non-IID settings, since even minor percentages of modified labels aggregate across iterations, significantly reducing global accuracy. Biggio et al. [22] presented data poisoning attacks, wherein malicious clients include samples with deliberately damaged labels or features into their local datasets, therefore modifying the statistical distribution of data and distorting the global model's representational efficacy. Fung et al. [14] highlighted feature perturbation attacks, when adversaries secretly manipulate feature values in local data, thus hiding discriminative patterns from the learning process and delaying convergence. These attacks rely on the reliance of FL on local updates without analyzing raw data, rendering malicious alterations challenging to identify at the server level.

2) Parameter-based Attacks

Bhagoji et al. [23] presented model poisoning attacks, in which adversaries systematically design malicious gradient updates to redirect global optimization towards attacker-specified objectives, therefore compromising model integrity. These kinds of attacks can be sneaky, preserving the accuracy on clean samples looking normal while adding specific weaknesses. Cheng et al. [24] examined gradient pollution attacks, wherein clients upload gradients that diverge from global objectives, so adding noise into the update process and causing convergence instability. Sun et al. [25] examined parameter forgery attacks, when malicious clients circumvent local training entirely and send falsified parameters that contradict real computations, so directly modifying the trajectory of the global model. More recently, model replacement backdoors have been proven to take advantage of scaling tactics that rewrite the global model in one round while keeping high clean accuracy. This shows how aggregation approaches that don't contain credibility checks aren't very useful.

B. Byzantine-Robust FL

In FL, Byzantine attackers can submit arbitrary malicious updates to servers, significantly jeopardising system security. To tackle this difficulty, researchers have proposed several defence strategies, primarily including distance-based approaches, statistical distribution-based approaches, and prior dataset-based approaches.

1) Distance-Based Approach

Huang et al. [21] presented Multi-Krum, a system that chooses client updates based on their proximity to neighboring clients in Euclidean space, nearly half of the participants being Byzantine. Multi-Krum performs effectively with IID data, but it has trouble in heterogeneous settings where honest gradients are spread out by nature. Fung et al. [14] presented Foolsgold, which mitigates the impact of sybil attacks by analyzing how similar gradients have been in the past and reducing the impact of clients with updates that are too closely related. However, this technique, on the other hand, requires a lot of computing power and might hurt good clients who have comparable responsibilities. Shejwalkar et al. [26] introduced Divide-and-Conquer (DnC), which divides clients into random subgroups and combines them independently to mitigate the effect of anomalies. However, the unpredictability of grouping creates instability, and the approach is still affected by how heterogeneous the client data is from one another. overall, distance-based defenses rely on benign updates are grouped tightly together. This assumption becomes much weaker in non-IID federated settings.

2) Statistical-based Approach

Yin et al. [27] proposed the Trimmed-Mean approach, which filter out a fixed proportion of extreme values from each coordinate before averaging them. This makes the technique more robust against a small number of adversaries. El Mhamdi et al. [28] improved this approach with Bulyan, a two-stage technique that integrates Multi-Krum for update preselection and Trimmed-Mean for final aggregation, providing enhanced assurances but incurring higher computing costs and necessitating more stringent assumptions on the quantity of benign clients. Pillutla et al. [17] proposed Robust Federated Aggregation (RFA), which utilizes the geometric median of updates to make the system strong against submissions that are very different or broken. Even though these strategies seem good in theory, they depend on the statistical assumption that harmless updates are clustered around a central mean. In heterogeneous federated environments, where benign clients may inherently provide divergent updates, these methods often misclassify honest participants as adversaries, resulting in the loss of vital information and a deceleration of convergence.

3) Prior Dataset-Based Approach

Cao et al. [18] presented FLTrust approach, which established trust on the server side by utilizing a small, clean dataset to calculate reference gradients and gives credibility weights to client updates based on cosine similarity. This approach works well to stop malicious influence, but it relies heavily on how well the reference dataset represents the real world. Park et al. [19] presented Sageflow, which integrates validation datasets and model verification procedures to collectively tackle adversarial updates and system failures. However, obtaining and maintaining validation data is costs and might not work well in sensitive areas like healthcare or finance. Huang et al. [20] developed SDEA, utilizing entropy features from public datasets to identify anomalous updates; yet, this method is susceptible to distribution discrepancies between proxy datasets and actual client populations. Prior dataset-based techniques have two significant challenges: the practical difficulty of acquiring high-quality labeled data and the bias created by dependence on public proxies, both of which restrict their generalizability in real-world federated settings.

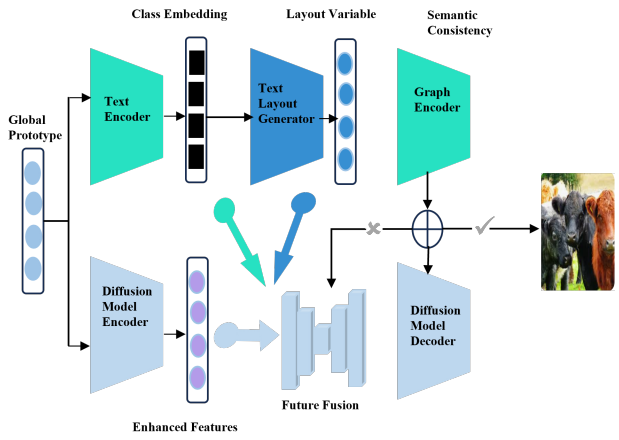

Fig. 1. Single Sample Generation Process

Algorithm 1: FedCWA

Input: Communication rounds T; client set K; number of clients U; class prototype collection $\{\mu_k^c \mid k \in K,\ c \in C\}$; diffusion-model encoder $f(\cdot)$ and decoder $G(\cdot)$; text encoder $f_{\text{text}}(\cdot)$; image encoder $f_{\text{image}}(\cdot)$; text-layout generator $g_{\text{TLG}}(\cdot)$; number of extended samples V; class set C; noise covariance matrix $\sigma^2\mathbf{I}$; diffusion-time parameter δ.

Output: Global model parameters w^{t+1}.

$w^0 \leftarrow$ randomly initialize global model parameters;
Server distributes w^0 to all clients;
for $k \in K$ ***in parallel*** **do**
  `// Generate class prototypes`
  $\{\mu_k^c \mid c \in C\} \leftarrow$ compute via Eq. (1);
  Upload $\{\mu_k^c \mid c \in C\}$ to server;
`// Generate fair proxy dataset`
$D_g \leftarrow$ ProDiff$(\{\mu_k^c\}, f, G, f_{\text{text}}, f_{\text{image}}, g_{\text{TLG}}, V, C, \delta, \sigma^2\mathbf{I})$;
for $t = 1, 2, \ldots, T$ **do**
  for $k \in K$ ***in parallel*** **do**
    $w_k^t \leftarrow$ LocalUpdating(w^t);
    Client k uploads w_k^t to the server;
  `// Aggregate local model parameters`
  $w^{t+1} \leftarrow$ CWA$(\{w_k^t \mid k \in K\}, D_g, K, U)$;
  Server distributes w^{t+1} to all clients;
return w^{t+1};

III. Proposed Detection Algorithm

A. Algorithms Overview

The proposed Byzantine-robust federated learning framework, FedCWA, as illustrated in Algorithm 1, operates through the following sequential components: 1) **Local prototype extraction.** Each client k in the client set K uses its local dataset to compute class prototypes $\{\mu_k^c \mid k \in K, c \in C\}$ and uploads them to the server; 2) **Global prototypes and proxy data.** The server aggregates these prototypes to obtain global class prototypes $\{\mu^c \mid c \in C\}$ and then employs the ProDiff method to generate a fair proxy dataset D_g; 3)**Credibility estimation and weight optimization.** The server computes the prediction credibility C_k for each client using the generated proxy dataset D_g together with the uploaded local model parameters w^t. It then dynamically optimizes the aggregation weights M_k via a Softmax over these credibility scores; 4) **Credibility-weighted aggregation.** The server aggregates local model parameters with the optimized weights M_k utilizing a weighted averaging scheme to obtain the new global model $w^{(t+1)}$; 5) **Broadcast.** The revised global model is distributed to all clients for the next round. After T rounds, the final model $w^{(t+1)}$ is secured.

B. Generation of Fair Proxy Dataset

The generation of the Fair Proxy Dataset (PDFD) is a crucial component of the FedCWA methodology. The generative pipeline for an individual sample is illustrated in Fig. 1, while

the full dataset building protocol is formalized in Algorithm 2. Unlike conventional approaches that rely on external prior datasets, FedCWA constructs a fair dataset directly from class prototypes uploaded by clients.

Here, class prototypes are class-centric features specifically, the mean feature vectors for each class. For samples of class c from client k, the prototype μ_k^c is computed as

$$\mu_k^c = \frac{1}{N_k^c} \sum_{(x,y)\in D_k^c} f(x) \tag{1}$$

Where D_k^c denotes all samples of class c on client k, N_k^c is the number of such samples, and $f(x)$ is the feature vector extracted from sample x by the feature extractor.

We employ a pre-trained diffusion model composed of an encoder $f(\cdot)$ and a decoder $G(\cdot)$. The encoder strengthens feature representations, and the decoder produces high-quality images. The server first receives the set of client prototypes $\{\mu_k^c \mid k \in K, c \in C\}$ and aggregates them to form global class prototypes:

$$\mu^c = \frac{1}{U} \sum_{k\in K} \mu_k^c \tag{2}$$

Where U is the number of participating clients contributing prototypes. The diffusion encoder then enhances each global prototype:

$$\mu^c = f(\mu^c) \tag{3}$$

With μ^c representing the enhanced class-feature embedding. In parallel, the server uses a text encoder $f_{\text{text}}(\cdot)$ to obtain a semantic embedding for each class, $\tau_c = f_{\text{text}}(c)$.

Next, latent features are sampled from the enhanced embeddings using Gaussian perturbation:

$$\tilde{z}_j^c \sim \mathcal{N}(\mu^c, \delta^2) \tag{4}$$

Where δ controls the perturbation magnitude.

To further improve generation quality, we adopt a hierarchical guidance mechanism via a text-layout generator $g_{\text{TLG}}(\cdot)$. It produces latent layout variables z_t, which are integrated into the reverse diffusion denoising process to guide image synthesis. The latent variables follow:

$$P_{\text{TLG}}(z_t \mid t) = \mathcal{N}(z_t; g_{\text{TLG}}(t), \sigma^2\mathbf{I}) \tag{5}$$

where $g_{\text{TLG}}(\cdot)$ is the latent layout function produced from text t, $\sigma^2\mathbf{I}$ is the noise covariance matrix, and $\mathbf{I}$ is the identity matrix.

Next, the server uses the diffusion decoder $G(\cdot)$ to synthesize images x'_j from latent features. During the reverse diffusion, the layout variables z_t are injected to refine the generated features:

$$\mu_\theta(x_t, z_t, t) = W_t \cdot \text{Concat}(x_t, z_t, t) \tag{6}$$

where $\text{Concat}(x_t, z_t, t)$ denotes concatenation of the noised image x_t, layout variable z_t, and text condition t; W_t is a learned weight matrix.

For each generated image, we obtain its semantic embedding z'_j via the image encoder $f_{\text{Image}}(\cdot)$ and compute its cosine similarity with the class text embedding τ_c:

$$\cos(\tau_c, z'_j) = \frac{\tau_c \cdot z'_j}{\|\tau_c\| \, \|z'_j\|} \tag{7}$$

Algorithm 2: ProDiff

Input: Class prototypes $\{\mu_k^c \mid k \in K,\ c \in C\}$; diffusion-model encoder $f(\cdot)$ and decoder $G(\cdot)$; text encoder $f_{\text{text}}(\cdot)$; image encoder $f_{\text{image}}(\cdot)$; text-layout generator $g_{\text{TLG}}(\cdot)$; number of extended samples V; class set C; noise covariance matrix $\sigma^2\mathbf{I}$; diffusion time parameter δ; semantic similarity threshold ε.

Output: Fair dataset D_g.

```
D_g ← ∅;
for c ∈ C do
    // Compute global class prototype
    μ^c ← calculate by Eq. (2);
    u^c ← calculate by Eq. (3);
    S_c ← ∅;
    τ_c ← f_text(c);
    for j = 1 to V do
        repeat
            z_j^c ← sample by Eq. (4);
            z_t ← sample by Eq. (5);
            // Generate layout variable
               and image
            μ_θ ← generate by Eq. (6);
            x'_j ← G(μ_θ);
            z'_j ← f_image(x'_j);
            cos(τ_c, z'_j) ← calculate by Eq. (12);
        until cos(τ_c, z'_j) > ε;
        S_c ← S_c ∪ {x'_j};
    D_g ← D_g ∪ S_c;
return D_g;
```

where $\cos(\tau_c, z'_j)$ measures the similarity between the class text embedding and the generated image embedding, and $\|\cdot\|$ is the Euclidean norm. If $\cos(\tau_c, z'_j) > \varepsilon$, the sample x'_j is accepted as semantically consistent and added to the extended set; otherwise, the perturbation process is repeated to regenerate the sample. Here, ε is the semantic-similarity threshold.

During training, two primary losses are optimized. First is the Semantic alignment loss L_{align}, this loss enforces consistency between the image embedding z'_j and the class text embedding τ_c in a shared semantic space. The alignment loss is formulated as:

$$L_{\text{align}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\cos(t, v_i)/\tau)}{\sum_{j=1}^{N} \exp(\cos(t, v_j)/\tau)} \tag{8}$$

where $\cos(t, v_i)$ is the cosine similarity between the class text t and the i-th image embedding v_i, τ is a temperature parameter, and N is the batch size. Minimizing L_{align} strengthens alignment between generated images and class semantics.

Second is the Diffusion loss L_{diff}, this loss trains the diffusion model to predict the injected noise so that synthesized images approach the real data distribution. The diffusion loss

is formulated as:

$$L_{\text{diff}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t, t)\|_2^2\right] \tag{9}$$

where ϵ is the noise added in the forward diffusion process, and $\epsilon_\theta(x_t, t, t)$ is the noise predicted by the reverse process. Minimizing L_{diff} enables recovery of high-quality images from noise.

Ultimately, the final loss function is derived by combining the alignment loss and the diffusion loss:

$$L = L_{\text{align}} + \lambda L_{\text{diff}} \tag{10}$$

where λ is a weighting coefficient that balances the two losses. Optimizing this joint objective yields semantically consistent, high-quality fair proxy datasets.

C. Credibility-Weighted Aggregation Method

In heterogeneous federated learning, the core objective of Byzantine-robust aggregation is to identify and mitigate the influence of malicious clients while amplifying the contribution of benign clients to the global model, thereby improving both fairness and robustness. To this end, we propose a Credibility-Weighted Aggregation (CWA) method that introduces a credibility notion to optimize client weights. The specific procedure is summarized in Algorithm 3, and the overall training flow is illustrated in Fig. 2.

To calculate the credibility C_k of each client, the server first uses the fair proxy dataset D_g generated on the server side. The server obtains each client's logits by combining the proxy samples $x_i \in D_g$ with the uploaded model parameters w_k. For each sample, applying w_k yields the logits vector represented as:

$$z_k(x_i) = [\, z_{k,1}(x_i), z_{k,2}(x_i), \ldots, z_{k,C}(x_i) \,] \tag{11}$$

where $z_k(x_i)$ is client k's original prediction logits for x_i, and the c-th component $z_{k,c}(x_i)$ is the model's prediction score for class c. For each proxy sample x_i, we take the prediction logits of client k and client j, denoted $z_k(x_i)$ and $z_j(x_i)$, and measure their similarity using cosine similarity:

$$S_{k,j}(x_i) = \frac{z_k(x_i) \cdot z_j(x_i)}{\|z_k(x_i)\| \cdot \|z_j(x_i)\|} \tag{12}$$

where "$\cdot$" denotes the vector dot product and $\|\cdot\|$ is the L_2 norm.

To comprehensively evaluate client k's overall similarity to others, we define an aggregated similarity S_k across all proxy samples and peer clients:

$$S_k = \frac{1}{|D_g|} \sum_{i \in D_g} \sum_{j \neq k} S_{k,j}(x_i) \tag{13}$$

Based on this similarity, the credibility of client k is defined as its average similarity with all other clients:

$$C_k = \frac{1}{U-1} \sum_{j \neq k} S_k \tag{14}$$

where U is the number of clients. After computing $\{C_k\}$, the server normalizes them to obtain dynamic aggregation weights. Clients with higher credibility receive larger weights, while those with lower credibility are down-weighted. The weights are computed via Softmax:

$$M_k = \frac{\exp(C_k)}{\sum_{j \in K} \exp(C_j)} \tag{15}$$

where M_k is the dynamic aggregation weight for client k, C_k is its credibility value, and K is the client set. Finally, the global model is updated by credibility-weighted averaging:

$$w^{(t+1)} = \sum_{k \in K} M_k \cdot w_k \tag{16}$$

The updated model $w^{(t+1)}$ is then broadcast to all clients for the next training round.

Algorithm 3: CWA

Input: Client model parameter set $\{w_k^t \mid k \in K\}$; fair proxy dataset D_g; client set K; number of clients U.

Output: Updated global model w^{t+1}.

```
for k ∈ K do
    for x_i ∈ D_g do
        // Compute prediction logits
        z_k(x_i) ← calculate by Eq. (11);
for k ∈ K do
    for x_i ∈ D_g do
        for j ≠ k do
            // Compute cosine similarity
            S_{k,j}(x_i) ← calculate by Eq. (12);
    S_k ← calculate by Eq. (13);
    // Compute credibility
    C_k ← calculate by Eq. (14);
// Compute dynamic weights and aggregate
M_k ← calculate by Eq. (15);
w^{t+1} ← calculate by Eq. (16);
return w^{t+1};
```

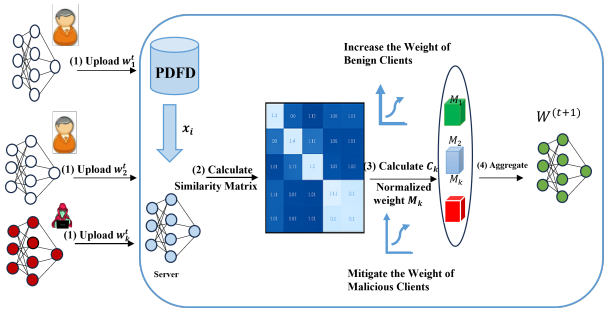

Fig. 2. CWA Method Framework

IV. Experimental Setup

A. Datasets and Model Architectures

We performed extensive experiments on three classical benchmark datasets to evaluate both the effectiveness and robustness of FedCWA:

- **CIFAR-10** [29]: Is a 10-class dataset which is balanced and includes 6,000 images per class, which covers natural objects, animals, and vehicles.
- **MNIST** [30]: 70,000 grayscale handwritten-digit images (28×28 pixels) spanning ten classes (0–9), divided into 60,000 training and 10,000 test samples.
- **Fashion-MNIST** [31]: 70,000 grayscale clothing images (28×28 pixels) in ten balanced categories.

For the model architecture, we employ SimpleCNN—a lightweight network with convolutional layers (3×3 kernels, ReLU), 2×2 max-pooling, batch normalization, and fully connected layers with Dropout—totaling approximately 1.2M parameters.

B. Proxy Dataset

To validate FedCWA's ability to generate fair proxy datasets, we used client-uploaded class prototypes with a diffusion model to synthesize unlabeled proxy data. For MNIST, we compared FedCWA-generated proxies with commonly used annotated proxy datasets:

- **USPS** [32]: 9,298 training and 2,007 test images (16×16 pixels) of handwritten digits. Although it shares the label space with MNIST, differences in collection environment and writing style introduce a notable domain shift.
- **SVHN** [33]: A considerable domain shift is evident in the 73,257 training and 26,032 test images (32×32 pixels) of house numbers from Google Street View, which are the result of real-world acquisition and complex backgrounds.
- **SYN** [34]: Is a dataset that has been algorithmically generated with deliberate distortions, rotations, and noise. It consists of 500,000 training and 100,000 test images.

C. Attack Setting

We validate robustness under two categories of Byzantine attacks.

1) Data poisoning attacks

- **Symmetric Label Flipping (SymF)** [35]: map each label y to $(y + 5)$ mod 10.
- **Paired Flipping (PairF)** [36]: frequent confusions (e.g., $3 \leftrightarrow 8$ for MNIST, cat↔dog for CIFAR-10). Noise rate for flipping is $\gamma = 0.5$ unless stated.

2) Model poisoning attacks

- **Random Noise (RanN)** [37]: add i.i.d. Gaussian noise to updates.
- **Little-Is-Enough (LIE)** [38]: craft small-magnitude adversarial updates to evade simple detectors.

D. Compared Defense Methods

We compare FedCWA with representative defenses from three families of common aggregation methods:

- **Distance-based algorithms**: including Multi-Krum, FoolsGold and DnC.
- **Statistical distribution-based schemes**: including Trimmed Mean, Bulyan and RFA.
- **Prior-dataset-based solutions**: including FLTrust, SageFlow and SDEA.

E. Implementation Details

1) **Training settings.** We set the number of communication rounds T to 100 or 50. This choice is based on empirical observations that model accuracy tends to plateau beyond these values, and thus further increasing T yields limited performance improvement while incurring additional computational overhead. The number of participating clients U is 10 or 20. For local training, we adopt FedProx as the optimization objective with 10 local update rounds. We use SGD as the local optimizer with an initial learning rate of 0.01, weight decay of 1×10^{-5}, and momentum of 0.9. The learning rate remains constant across rounds because we empirically found that a fixed learning rate leads to more stable convergence under heterogeneous client updates. In FedCWA, the fair-dataset mini-batch size is 64, trained with Adam (learning rate $\eta = 0.005$) for 20 epochs, which balances convergence speed and stability in auxiliary-model learning. 2) **Attack settings.** The Byzantine attack ratio Φ is 0.2. The noise rate γ for Symmetric Flipping and Paired Flipping is set to 0.5 by default. 3) **Data heterogeneity.** We use a Dirichlet distribution Dir(β) to simulate label-skewed non-IID data. The parameter $\beta > 0$ controls the level of label imbalance: smaller β values correspond to more heterogeneous client distributions. Following common practice in federated learning literature and to capture moderate and strong heterogeneity levels, we set $\beta = 0.5$ and $\beta = 0.3$ in our experiments. 4) **Evaluation metric.** We report Top-1 accuracy. The final results are obtained by averaging the accuracy of the last five communication rounds to reduce randomness in late-stage fluctuations.

V. Experimental Results

We evaluate FedCWA against state-of-the-art Byzantine-robust aggregation methods on CIFAR-10, MNIST, and Fashion-MNIST under diverse attack scenarios and non-IID conditions. We focus on three objectives: model accuracy, convergence behavior, and malicious-client identification.

A. Model Accuracy

Tables I, II, and III present the model accuracy comparison accross different datasets and Byzantine attacks types.

On CIFAR-10 (Table I), FedCWA achieves 67.23% accuracy under SymF attack with $\beta = 0.5$, outperforming Multi-Krum by 9.52 percentage points. When data heterogeneity increases ($\beta = 0.3$), FedCWA maintains stable performance (67.24%), demonstrating its robustness to data distribution shifts.

On MNIST (Table II), FedCWA demonstrates exceptional robustness against PairF attacks, maintaining 99.43% accuracy while Multi-Krum drops to 11.35%. Under RanN attacks, FedCWA achieves 99.27% accuracy, outperforming Multi-Krum (83.13%) by 16.14 percentage points.

On Fashion-MNIST (Table III), FedCWA achieves 88.82% accuracy under SymF attacks, significantly outperforming Multi-Krum (10%) by 78.82 percentage points. When data heterogeneity increases ($\beta = 0.3$), FedCWA maintains stable performance across all attack types with variations less than 0.25%, confirming its stability under different data distributions.

Tab. I. Performance comparison of different methods on CIFAR-10 dataset

Method	$\beta = 0.5$				$\beta = 0.3$			
	PairF	SymF	RanN	LIE	PairF	SymF	RanN	LIE
Multi Krum	52.07	57.71	60.45	62.04	51.80	51.42	52.92	51.46
Bulyan	46.79	44.02	51.83	54.25	29.30	36.06	37.76	48.23
Trim Median	48.91	51.23	53.70	50.09	47.51	49.01	48.06	54.34
FoolsGold	60.05	60.69	62.43	63.86	54.58	59.42	60.40	45.88
DnC	65.55	64.72	64.72	64.21	65.56	63.02	64.54	64.06
RFA	67.12	64.17	58.19	64.84	**66.66**	63.93	56.81	59.91
SDEA	67.68	65.82	65.21	68.42	66.23	66.72	**67.27**	68.32
FedCWA	**68.91**	**67.23**	**69.25**	**68.59**	66.65	**67.24**	67.09	**68.55**

Tab. II. Performance comparison of different methods on MNIST dataset

Method	$\beta = 0.5$				$\beta = 0.3$			
	PairF	SymF	RanN	LIE	PairF	SymF	RanN	LIE
Multi Krum	11.35	87.83	83.13	11.35	9.87	81.72	9.70	11.35
Bulyan	96.69	97.64	98.90	98.86	98.48	98.40	99.05	99.14
Trim Median	96.21	95.80	96.97	98.12	97.10	97.41	98.04	98.35
FoolsGold	98.32	98.73	98.88	98.63	98.71	98.71	98.69	98.66
DnC	98.72	99.22	98.61	99.21	98.40	98.53	98.52	98.58
RFA	98.75	98.34	98.23	98.91	98.23	98.41	98.63	98.54
SDEA	98.78	**99.03**	98.94	98.79	98.46	98.84	98.61	**99.04**
FedCWA	**99.43**	98.96	**99.27**	**99.32**	**99.12**	**98.88**	**99.26**	**99.04**

B. Convergence Performance

To analyze model convergence, we selected all typical attack scenarios and plotted the average accuracy curves during training. As shown in Figure 3, FedCWA demonstrates excellent convergence properties under both attacks: under SymF attacks, the model stabilizes after approximately 50 communication rounds with a final accuracy of 65.82%; under RanN attacks, the model converges faster, reaching 69.21% accuracy in just 40 rounds. These results confirm FedCWA's robustness and stability across different attack types.

C. Impact of Proxy Dataset Selection

Table IV compares methods using different proxy datasets. In contrast to existing approaches, FedCWA exhibits superior classification accuracy through its PDFD mechanism, effectively capturing heterogeneous distribution characteristics without reliance on prior datasets. Empirical evidence demonstrates that FedCWA maintains high prediction precision across multiple non-IID scenarios and adversarial attack vectors.

VI. Conclusion

In this paper, we propose FedCWA, a novel federated learning defense that detects and mitigates malicious clients by generating fair proxy datasets. FedCWA leverages diffusion models and client class prototypes to synthesize high-quality unlabeled proxy data, and combines them with a credibility-weighted aggregation strategy to effectively identify and suppress malicious updates. Extensive experiments on CIFAR-10, MNIST, and Fashion-MNIST demonstrate the strong robustness of FedCWA under diverse attack methods; across datasets and scenarios, it consistently outperforms existing defenses. While FedCWA provides a promising step toward secure federated learning, several open challenges remain. Future work includes integrating stronger privacy-preserving mechanisms (e.g., differential privacy or secure aggregation) into the FedCWA pipeline to further balance security and privacy. In addition, extending our proxy-data generation mechanism to more complex data modalities and exploring its applicability in cross-device or large-scale heterogeneous settings represent valuable directions. We also plan to investigate adaptive attacks targeting synthetic proxy data generation, enabling a more comprehensive evaluation of defense resilience.

Tab. III. Performance comparison of different methods on Fashion-MNIST dataset

Method	$\beta = 0.5$				$\beta = 0.3$			
	PairF	SymF	RanN	LIE	PairF	SymF	RanN	LIE
Multi Krum	10.00	10.00	75.00	10.00	36.03	45.31	10.12	10.00
Bulyan	84.35	85.33	87.53	87.05	82.34	81.41	86.04	86.52
Trim Median	84.11	85.21	86.82	86.64	75.14	75.86	81.72	83.43
FoolsGold	61.52	43.96	55.89	71.79	72.99	60.76	61.50	72.03
DnC	87.65	10.00	87.09	87.03	86.04	85.92	86.60	86.76
RFA	88.24	88.39	87.45	87.73	87.21	88.62	87.90	87.68
SDEA	87.74	87.75	**88.20**	**88.29**	88.21	88.26	87.93	88.22
FedCWA	**88.66**	**88.82**	87.97	88.27	**88.49**	**88.27**	**87.94**	**88.26**

Tab. IV. Performance comparison under different proxy datasets ($\beta = 0.5$)

Dataset	Method	PairF	SymF	RanN	LIE
USPS	FLTrust	11.35	70.21	11.35	36.20
	Sageflow	98.88	98.08	99.16	99.10
	SDEA	99.05	99.01	99.13	**99.15**
	FedCWA	**99.15**	**99.20**	**99.25**	99.12
SVHN	FLTrust	79.80	85.11	72.79	11.35
	Sageflow	99.12	98.78	99.08	99.09
	SDEA	99.10	**99.27**	99.15	99.12
	FedCWA	**99.16**	**99.27**	**99.21**	**99.21**
SYN	FLTrust	65.81	83.14	78.30	96.69
	Sageflow	99.16	92.57	99.06	99.13
	SDEA	99.14	99.01	99.08	**99.15**
	FedCWA	**99.18**	**99.08**	**99.13**	99.14
PDFD	FLTrust	70.00	80.00	75.00	90.00
	Sageflow	99.00	98.00	99.00	99.00
	SDEA	99.10	**99.03**	99.03	99.10
	FedCWA	**99.20**	99.00	**99.22**	**99.20**

References

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

[2] Y. Shi, H. Song, and J. Xu, "Responsible and effective federated learning in financial services: A comprehensive survey," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 4229–4236.

[3] S. T. Ahmed, A. C. Kaladevi, K. V, A. Shankar, and F. Alqahtani, "Privacy enhanced edge-ai healthcare devices authentication: A federated learning approach," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2025.

[4] T. Zheng, A. Li, Z. Chen, H. Wang, and J. Luo, "Autofed: Heterogeneity-aware federated multimodal learning for robust autonomous driving," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, ser. ACM MobiCom '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3570361.3592517

[5] P. R. Ovi and A. Gangopadhyay, "Robust federated learning against data poisoning attacks: Prevention and detection of attacked nodes," *Electronics*, vol. 14, no. 15, 2025. [Online]. Available: https://www.mdpi.com/2079-9292/14/15/2970

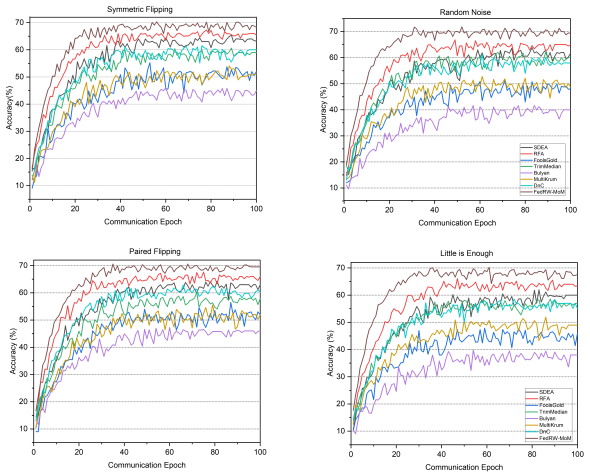

Fig. 3. Model Convergence performance under different attack scenario

[6] L. Yang, Y. Miao, Z. Liu, Z. Liu, X. Li, D. Kuang, H. Li, and R. H. Deng, "Enhanced model poisoning attack and multi-strategy defense in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 3877–3892, 2025.

[7] W. Shen, W. Huang, G. Wan, and M. Ye, "Label-free backdoor attacks in vertical federated learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 19, pp. 20 389–20 397, Apr. 2025. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/34246

[8] X. Mu, K. Cheng, T. Liu, T. Zhang, X. Geng, and Y. Shen, "Fedpta: Prior-based tensor approximation for detecting malicious clients in federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 9100–9114, 2024.

[9] H. Xiao, X. Mu, and K. Cheng, "Fedrma: a robust federated learning resistant to multiple poisoning attacks," *Journal of Networking and Network Applications*, vol. 4, no. 1, pp. 31–38, 2024.

[10] X. Mu, K. Cheng, Y. Shen, X. Li, Z. Chang, T. Zhang, and X. Ma, "Feddmc: Efficient and robust federated learning via detecting malicious clients," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 6, pp. 5259–5274, 2024.

[11] A. Song, T. Zhang, K. Cheng, Y. Cao, X. Zhu, and Y. Shen, "Byzantine-robust federated learning framework via a server-client defense mechanisms," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 29 073–29 088, 2025.

[12] H. Kabbaj, R. El-Azouzi, and A. Kobbane, "Robust federated learning via weighted median aggregation*," in *2024 2nd International Conference on Federated Learning Technologies and Applications (FLTA)*, 2024, pp. 298–303.

[13] L. Huo, L. Wu, J. Feng, X. Fan, E. Wang, and X. Li, "Dp-caka: Defending local model poisoning attacks based on differential privacy and complex acc-based multi-krum algorithm in distributed federated learning," in *2024 IEEE International Conference on High Performance Computing and Communications (HPCC)*, 2024, pp. 1409–1418.

[14] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2020. [Online]. Available: https://arxiv.org/abs/1808.04866

[15] T. Wang, Z. Zheng, and F. Lin, "Federated learning framework based on trimmed mean aggregation rules," *Expert Systems with Applications*, vol. 270, p. 126354, 2025.

[16] S. Pandey, O. Singh, A. Pandey, and C. Pandey, "Robust and privacy-preserving federated learning against malicious clients: A bulyan-based adaptive differential privacy framework," *IEEE Access*, vol. 13, pp. 139 931–139 943, 2025.

[17] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1142–1154, 2022.

[18] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," 2022. [Online]. Available: https://arxiv.org/abs/2012.13995

[19] J. Park, D.-J. Han, M. Choi, and J. Moon, "Sageflow: Robust federated learning against both stragglers and adversaries," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 840–851.

[20] W. Huang, Z. Shi, M. Ye, H. Li, and B. Du, "Self-driven entropy aggregation for byzantine-robust heterogeneous federated learning," in *Forty-first International Conference on Machine Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=k2axqNsVVO

[21] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, ser. AISec '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 43–58. [Online]. Available: https://doi.org/10.1145/2046684.2046692

[22] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," 2013. [Online]. Available: https://arxiv.org/abs/1206.6389

[23] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 634–643. [Online]. Available: https://proceedings.mlr.press/v97/bhagoji19a.html

[24] R. Cheng, X. Wang, F. Sohel, and H. Lei, "Topology-aware universal adversarial attack on 3d object tracking," *Visual Intelligence*, vol. 1, no. 1, p. 31, 2023. [Online]. Available: https://doi.org/10.1007/s44267-023-00033-8

[25] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" 2019. [Online]. Available: https://arxiv.org/abs/1911.07963

[26] V. Shejwalkar and A. Houmansadr, "Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning," in *NDSS*, 2021.

[27] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a.html

[28] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," 2018. [Online]. Available: https://arxiv.org/abs/1802.07927

[29] A. Krizhevsky, G. Hinton *et al.*, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, no. 7, pp. 1–9, 2010.

[30] S. Peng, Y. Yang, M. Mao, and D.-S. Park, "Centralized machine learning versus federated averaging: A comparison using mnist dataset." *KSII Transactions on Internet & Information Systems*, vol. 16, no. 2, 2022.

[31] J. Reyes, L. Di Jorio, C. Low-Kam, and M. Kersten-Oertel, "Precision-weighted federated learning," *arXiv preprint arXiv:2107.09627*, 2021.

[32] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, A. Y. Ng *et al.*, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, vol. 2011, no. 5. Granada, 2011, p. 7.

[34] P. Roy, S. Ghosh, S. Bhattacharya, and U. Pal, "Effects of degradations on deep neural network architectures," 2025. [Online]. Available: https://arxiv.org/abs/1807.10108

[35] B. van Rooyen, A. Menon, and R. C. Williamson, "Learning with symmetric label noise: The importance of being unhinged," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[36] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Robust training of deep neural networks with extremely noisy labels," in *Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS)*, vol. 2, 2020, p. 4.

[37] J. Shi, W. Wan, S. Hu, J. Lu, and L. Yu Zhang, "Challenges and approaches for mitigating byzantine attacks in federated learning," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2022, pp. 139–146.

[38] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.