

Securing Federated Learning: a comprehensive survey on poisoning attacks and defense mechanisms

Xiaohu Shan^{1#}, Jianyao Zhang^{1#}, Mingying Zhang^{1,2*}, Haiyang Yu¹ and Zhen Yang¹

¹ School of Computer Science, Beijing University of Technology, Beijing 100124, China

² China Electronics Standardization Institute, Beijing 100007, China

Authors contributed equally: Xiaohu Shan, Jianyao Zhang

* Correspondence: zhangmingying@emails.bjut.edu.cn (Zhang M)

Abstract

Federated Learning (FL) enables privacy-preserving collaborative training by keeping sensitive data localized. However, the decentralized nature of FL prevents the central server from verifying local training processes, leaving the system vulnerable to poisoning attacks. In these scenarios, Byzantine clients upload malicious local gradients or model updates to either degrade the global model's performance (untargeted attacks) or inject backdoor triggers (targeted attacks). As FL applications scale, both attack vectors and defensive countermeasures are rapidly evolving. This survey provides a comprehensive analysis of the current FL security landscape. Specifically, we: (1) conduct a systematic technical analysis of existing poisoning attack methods, with an emphasis on their objectives, optimization strategies, and stealth characteristics; (2) summarize and critically compare mainstream defense strategies from multiple perspectives, including robustness, adaptability, and applicability under non-IID settings; (3) explore the adversarial co-evolution between attacks and defenses to reveal their underlying design principles; and (4) identify open challenges, emphasizing the critical trade-offs between privacy-preserving techniques, model robustness, and communication efficiency. This paper aims to provide a structured reference and a unified perspective for researchers and practitioners in secure federated learning.

Keywords: Federated learning, Poisoning attacks and defense, Privacy-preserving

Citation: Shan X, Zhang J, Zhang M, Yu H, Yang Z. 2026. Securing Federated Learning: a comprehensive survey on poisoning attacks and defense mechanisms. *The Knowledge Engineering Review* 41: e014 <https://doi.org/10.48130/ker-0026-0014>

1 Introduction

In today's data-driven era, artificial intelligence (AI) technology is advancing at a rapid pace, driving the implementation and application of numerous intelligent services—with personalized recommendation systems being one of the most representative examples^[1]. However, machine learning (ML), the core technology underlying AI, is confronted with severe challenges such as privacy leakage and trust crises^[2]. Against the backdrop of users' growing focus on data security, they tend to store data locally, which makes data circulation difficult and gives rise to the so-called "data silos"^[3,4]. The phenomenon of data silos severely hinders cross-entity data collaboration and prevents the full release of the potential value of data.

To address this issue, Google proposed Federated Learning (FL), an innovative distributed machine learning framework in 2016. FL enables multiple participants to collaboratively train a model without sharing their raw data. Each user's data remains locally stored at all times, which not only safeguards the ownership and control rights of the data but also mitigates the privacy risks that may arise from centralized data management^[5]. By conducting model training locally and only uploading parameter updates, FL effectively alleviates the problem of data silos, thereby enhancing model performance while achieving the protection of user privacy^[6].

Owing to the characteristics of FL, such as distribution, large scale, and high client autonomy^[7], FL systems are highly vulnerable to poisoning attacks during the training process^[8,9]. Malicious clients undermine the global model training by uploading forged or tampered local training results^[10,11]. The attackers' objectives may include degrading the overall performance of the model, increasing prediction

errors, or causing the model to malfunction on specific tasks. Such Byzantine attacks typically exhibit strong disruptive effects and are difficult for traditional aggregation strategies to detect and resist in a timely manner^[12]. In contrast, backdoor attacks are more concealed. Attackers inject elaborately designed "poisoned samples" into local datasets, enabling the model to output incorrect predictions under specific trigger conditions^[13]. For instance, when training a traffic sign recognition model, an attacker may label a "stop sign" as a "speed limit sign" and embed a specific trigger pattern into the sample. After training, the model will be activated to execute the backdoor and make erroneous judgments when it encounters similar patterns^[14]. Unlike Byzantine attacks, backdoor attacks rarely impair the model's performance on normal data; thus, they are more difficult to detect and pose higher security risks^[15–17].

Although FL exhibits certain advantages in data privacy protection, issues such as information leakage and expanded attack surfaces persist during the process of model parameter sharing and aggregation^[18]. Attackers can even infer the private content of users' local data by analyzing the gradient information in model updates, which further amplifies security threats. Therefore, the development of detection and defense mechanisms against these two types of poisoning attacks has become a critical research direction for ensuring the security and trustworthiness of FL systems^[19].

2 Notation

To improve readability, Table 1 summarizes the frequently used symbols in this survey. Since this paper reviews methods from different studies, some equations preserve the notation used in the original

Table 1. Frequently used notation in this survey.

Symbol	Description
t	Communication round in federated learning.
i, k	Client index.
K	Total number of clients.
m	Number of malicious clients or fake clients, depending on the context.
w^t, W_t , or G_t	Global model at round t , with notation depending on the reviewed method.
w_i^t or W_i^t	Local model or local update submitted by client i .
Δ_i^t or g_i^t	Local model update or gradient update of client i .
$\mathcal{A}(\cdot)$ or $\Lambda(\cdot)$	Aggregation rule or aggregation process.
s	Attack direction or estimated benign update direction.
S	Context-dependent symbol; it may denote an output event set in DP definitions or a clipping bound in DP-SGD.
ϵ, δ	Differential privacy parameters.
σ	Gaussian noise scale or noise multiplier.
λ, τ, β	Method-specific control parameters.
ASR	Attack success rate.
MTA	Main task accuracy.

works. When method-specific symbols are introduced, we provide their meanings in the corresponding context.

3 Analysis of poisoning attacks in federated learning

Poisoning attacks, commonly used in the field of machine learning, refer to a type of malicious method whose primary goal is to impair the performance of machine learning models^[20]. They attempt to disrupt the model's learning process by controlling and manipulating part of the training data or local models, thereby leading to incorrect prediction or classification results^[21]. FL systems are vulnerable to poisoning attacks launched by adversaries for the following reasons: (1) In FL, training data is distributed across multiple participants, which complicates the monitoring and maintenance of data quality. Malicious participants can easily inject fake or contaminated data, as data management is decentralized among multiple parties. (2) Participants are required to exchange information with the server to collaboratively train the model^[22]. The exchanged information includes gradient updates, model parameters, and so on. Malicious participants can submit poisoned information during this process, thereby compromising the performance of the global model. (3) To protect personal privacy, participants usually adopt privacy-preserving measures when submitting model updates^[23]. Such schemes make it difficult for the server to detect poisoned data from ciphertext data. In general, poisoning attacks can be classified into two categories: untargeted poisoning attacks (Byzantine attacks) and targeted poisoning attacks (backdoor attacks), as summarized in Table 2. In the subsequent sections, the latest research on these two types of poisoning attacks will be presented^[24].

3.1 Byzantine attacks

At a high level, Byzantine attacks aim to corrupt the global optimization process by manipulating the model updates submitted by compromised clients. Unlike backdoor attacks, which usually preserve clean accuracy while injecting hidden malicious behavior, Byzantine attacks mainly target the availability and convergence of the global model. Recent Byzantine attacks have evolved from simple perturbation-based strategies to more structured attacks that exploit

contribution evaluation, aggregation-rule geometry, or client participation mechanisms.

As shown in Table 3, Byzantine attacks refer to adversarial behaviors where attackers, whether acting proactively or passively, compromise the global model by controlling or manipulating one or more local clients. Specifically, attackers achieve this by tampering with the datasets held by the compromised clients or altering the gradient parameters uploaded during the training process, with the core goal of degrading model performance^[10]. Byzantine attacks aim to undermine training availability and model availability: they prevent the global model from converging or attaining optimal performance on the primary training task, and do not target any specific users or data samples. Based on the latest existing literature, we have synthesized three novel Byzantine attack methods in FL^[19]. The specific details are shown in Table 3.

3.1.1 Contribution evaluation poisoning targeting the FL workflow

This type of attack focuses on vulnerabilities in the core workflow between the contribution-evaluation and the model-aggregation in FL, and achieves stealthy poisoning by manipulating the apparent contribution of local model updates. The ACE attack^[25] proposed by Xu et al. serves as a typical representative of this category. Its core mechanism lies in the fact that servers typically measure client contributions using the similarity between local models and the global model. Malicious clients can construct local updates that are highly aligned with the global model by predicting the direction of the future global model, thereby disguising themselves as high-contribution nodes. Its attack principle is shown in Fig. 1.

Specifically, the ACE attack implements its strategy through two core components: The first component predicts the future global model update based on the Cauchy Mean Value Theorem and the L-BFGS algorithm, and reduces computational complexity using historical global model snapshots. According to the Cauchy Mean Value Theorem, the formula for predicting the global model update $\hat{g}^t$ at the t -th round is expressed as follows:

$$\hat{g}^t = g^{t-1} + H^t(w^t - w^{t-1}) \quad (1)$$

Table 2. Comparison of Byzantine and backdoor attacks in federated learning.

Category	Byzantine attacks	Backdoor attacks
Objective	Global performance degradation	Hidden malicious functionality
Target	Training and aggregation	Inference behavior
Specificity	Untargeted	Targeted
Attack means	Malicious model updates	Poisoned data or updates
Clean accuracy	Severely affected	Largely preserved
Stealthiness	Low	High
Activation	Always active	Trigger-based
Threat type	Availability and robustness	Integrity and trustworthiness

Table 3. Comparison of poisoning attack methods in FL.

Attack	Mechanism	Stealthiness
ACE	Predict global update, disguise contribution	High
OPT	Reverse optimization vs aggregation rules	Moderate
MPAF	Large-scale fake clients push base model	Moderate
PF	Flip critical bits in deployed model	High
3DF	Multi-objective optimization + noise masks	Very high

ACE: contribution evaluation poisoning; OPT: optimization-based poisoning; MPAF: fake-client poisoning; PF: ProFlip bit-flip backdoor; 3DF: adaptive backdoor.

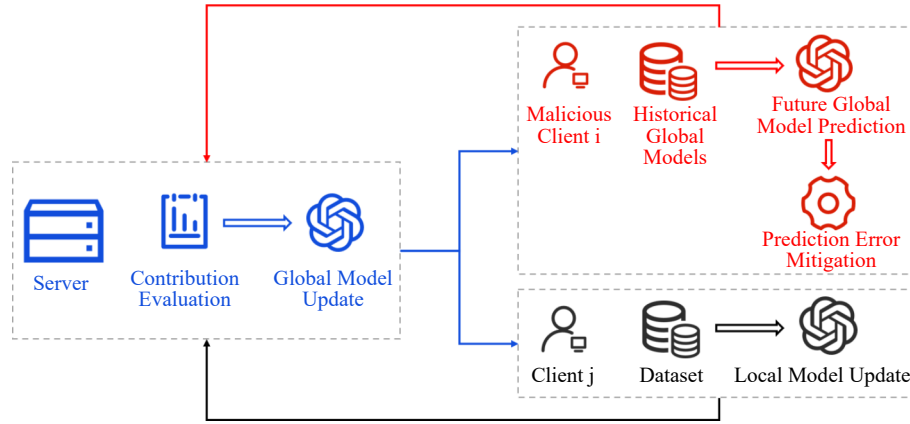


Fig. 1 Workflow of contribution-evaluation poisoning attack in federated learning.

where $\hat{g}^t$ denotes the global model update from the previous round, W_t and W_{t-1} represent the current and previous global models, respectively, and H^t stands for the integrated Hessian matrix.

The second component mitigates the accumulation of prediction errors through a threshold filtering strategy, with the judgment condition defined as:

$$\|L - BFGS(\Delta W^t, \Delta G^t, w^t - w^{t-1})\| \leq \tau, (\tau = \|w^t - w^{t-1}\|, l = 1) \quad (2)$$

If this condition is not satisfied, the attack switches to a benign update mode. For scenarios where contribution is evaluated using cosine distance, the optimization objective is formulated as: $\{\hat{g}_i^t | i \in \hat{\Gamma}\} = \arg\max_{\{g_i | i \in \Gamma\}} \sum_{i \in \hat{\Gamma}} (1 - \cos(g_i, g^t))$, and the effectiveness can be enhanced by introducing an amplification factor c .

Experimental results demonstrate that this attack can successfully deceive five mainstream contribution evaluation methods, including FedSV and GDR. In non-independent and identically distributed (non-IID) data scenarios, it outperforms baseline attacks in both contribution score and ranking gains, while requiring extremely low computational and storage costs. Meanwhile, the fluctuation in the test accuracy of the global model is controlled within 1%, perfectly meeting the stealth requirement of "undermining contribution fairness without revealing attack intentions". Notably, the ACE attack is independent of aggregation rules and exhibits strong generality; existing defense measures such as Multi-Krum and Trimmed Mean fail to detect or resist it effectively^[26].

3.1.2 Model poisoning attack based on objective optimization

In FL, this type of attack maximizes the deviation of the global model as its optimization objective and reversely constructs malicious updates by targeting the mathematical properties of Byzantine-resilient aggregation rules. The local model poisoning attack^[27] proposed by Fang et al. is the first to systematically implement this strategy: attackers compromise a subset of clients and manipulate local model parameters during training, leading to a significant increase in the test error of the global model. This attack paradigm formulates the attack problem as an optimization task that "maximizes the deviation of the global update from the benign direction under the constraints of aggregation rules". Its core optimization objective is expressed as follows:

$$\max_{w_1', \dots, w_c'} s^T (w - w') \quad (3)$$

$$s.t. w = A(w_1, \dots, w_m), w' = \Lambda(w_1', \dots, w_m') \quad (4)$$

where s denotes the direction vector of changes in global model parameters in the absence of attacks; w (or w') represents the global model

without (or with) attacks; Λ stands for the aggregation rule; and w_i (or w_i') denotes the benign (or malicious) local model. This formulation shows that the attacker does not simply inject arbitrary noisy updates, but constructs malicious local models according to the behavior of the target aggregation rule. The term $s^T (w - w')$ measures how much the poisoned global model deviates from the benign update direction. A larger value indicates that the attacker has pushed the aggregation result further away from the expected benign trajectory. Therefore, the effectiveness of optimization-based poisoning largely depends on the attacker's ability to estimate the benign direction and exploit the decision boundary of the aggregation rule. Attack strategies vary with different aggregation rules. For the Krum rule (which filters local models based on pairwise distances between them), attackers utilize a "support cluster" mechanism to make multiple malicious clients send similar forged updates, forcing the server to select poisoned models. The core lies in constructing malicious models that satisfy the conditions: $w_1' = w_{Re} - \lambda s$ and $w_i' = w_1'$ (for $i = 2, \dots, c$), where w_{Re} is the global model from the previous round, and $\lambda > 0$ is the deviation coefficient. For the Trimmed Mean and Median rules (which adopt coordinate-wise aggregation), attackers construct malicious parameters close to extreme values, thereby shifting the trimmed mean/median toward the malicious direction. Specifically, if $s_j = -1$ (indicating that the j -th dimension parameter needs to be increased), c malicious parameters larger than $w_{max,j}$ (the maximum value of benign parameters in the j -th dimension) are constructed; otherwise, parameters smaller than $w_{min,j}$ (the minimum value of benign parameters in the j -th dimension) are constructed.

3.1.3 Efficient model poisoning attack based on fake clients

To overcome the cost constraint of compromising a large number of real clients, this type of attack achieves efficient poisoning by forging low-cost virtual clients to send malicious updates in bulk. The MPAF attack^[28] proposed by Cao & Gong subverts the resource dependence of traditional attacks: attackers can generate a large number of fake clients using tools such as Android emulators and launch attacks without any real local data. Its attack principle is shown in Fig. 2.

The attack logic of MPAF is as follows: a randomly initialized model with low accuracy is selected as the "base model". The core lies in formulating the attack objective as an optimization problem that drives the final global model to approximate this base model:

$$\min_{g_i^t, i \in [n+1, n+m], t \in [0, T-1]} \|w^T - w'\| \quad (5)$$

where w' denotes the base model, w^T represents the final global model, n (or m) stands for the number of real (or fake) clients, and T is the number of training rounds. In each iteration round, each fake client constructs an update in the direction of "base model-current global

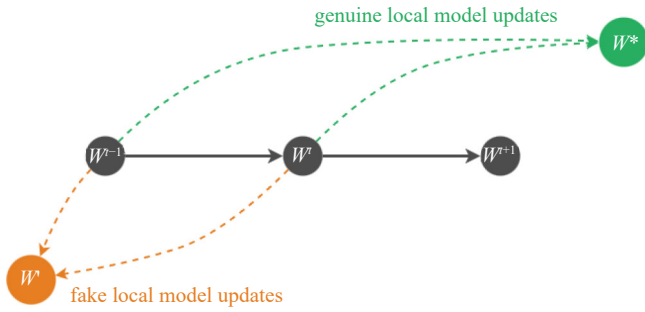


Fig. 2 Workflow of fake-client-based model poisoning attack in federated learning.

model" and enhances its weight in aggregation via an amplification factor λ , continuously pulling the global model toward the base model. The formula for constructing the fake local update is expressed as follows.

$$g'_t = \lambda(w' - w^t) \quad (6)$$

where w' is the global model in the t -th round, and the default value of λ is 1×10^6 .

These three types of Byzantine attacks have evolved along three distinct dimensions: process mechanism exploitation, mathematical optimization breakthroughs, and resource cost reduction. All of them take model poisoning as their core means, yet differ in their attack entry points and stealthiness. Specifically, process-targeted attacks focus on the undermining of contribution fairness; optimization-based attacks emphasize the effect of model performance collapse; and fake-client-based attacks highlight large-scale attack efficiency^[29].

3.2 Backdoor attacks

3.2.1 Backdoor attack based on bit flipping

The core of this type of attack lies in directly modifying the binary representation of the parameters stored in the memory after the model is deployed, which means it could alter the behavior of the model^[30]. This kind of attack completely bypasses the monitoring during the training process.

For example, ProFlip^[31] launches runtime attacks through physically flipping bits after the model is deployed. Its objective is to find the smallest set of bit flips in the model parameters so that when the input contains a specific trigger, the model will be misled to the target category. Its workflow is shown in Fig. 3. ProFlip employs a progressive bit-searching strategy. Firstly, it identifies and locates the neurons that are important for the target category through adversarial saliency maps. Then, the trigger generation process is formalized as an optimization problem:

$$\mathcal{L}_{\text{trig}} = \lambda_1 \cdot \mathcal{L}_{\text{mse}}(M_{1:L-1}(A(\mathbf{x}, \mathbf{m}, \Delta)); c) + \lambda_2 \cdot \mathcal{L}_{\text{ce}}(M(A(\mathbf{x}, \mathbf{m}, \Delta)); t), \quad (7)$$

where $A(\cdot)$ applies the trigger, $\mathcal{L}_{\text{mse}}$ pushes activations of salient neurons towards a large constant c , and $\mathcal{L}_{\text{ce}}$ causes misclassification to the target class.

The core of ProFlip is its *Critical Bit Search (CBS)* algorithm. It gradually identifies the most vulnerable bits by first selecting a parameter element that maximizes a *fitness score* F :

$$F(p, elem) = \left| \frac{\partial \mathcal{L}_{\text{CBS}}}{\partial p} \right|_{elem} \cdot step. \quad (8)$$

The gradient magnitude $|\partial \mathcal{L}_{\text{CBS}} / \partial p|$ measures the sensitivity of the loss to changes in the element $elem$, and $step$ is the maximum allowable change for that element within its quantized range. This score prioritizes elements that are highly sensitive and can be perturbed significantly.

Once the most vulnerable element ($p_{\text{sens}}, elem_loc$) is found, ProFlip determines the optimal value for this element by minimizing the Trojan loss:

$$\mathcal{L}_{\text{troj}} = \gamma_1 \cdot \mathcal{L}_{\text{ce}}(M^*, \mathcal{D}) + \gamma_2 \cdot \mathcal{L}_{\text{CBS}}. \quad (9)$$

Here, $\mathcal{L}_{\text{ce}}(M^*, \mathcal{D})$ ensures the Trojaned model M^* maintains accuracy on clean data $\mathcal{D}$, preserving stealth. The optimal value is found via a grid search over the quantized value range. The number of bit flips required n_b is the Hamming distance between the original and new binary representations of the element. This greedy, progressive search from parameter to element to bit value allows ProFlip to achieve a high Attack Success Rate (ASR) with an astonishingly small n_b (e.g., 12 bits for ResNet-18 on CIFAR-10), making it a potent and practical threat.

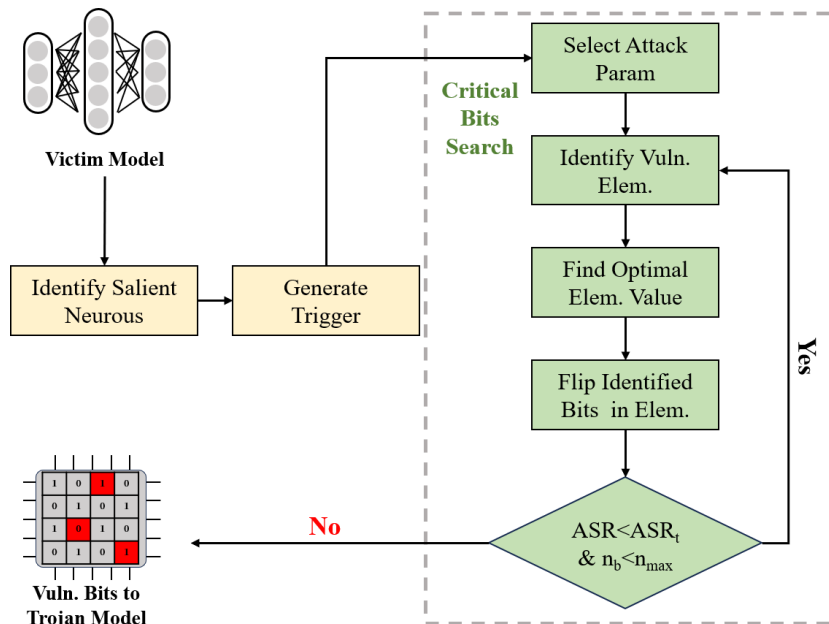


Fig. 3 ProFlip workflow.

3.2.2 Adaptive attack based on multi-objective optimization

In FL, malicious clients aim to poison the global model by submitting malicious updates that introduce a backdoor functionality, which is activated only by a specific trigger pattern during inference. The primary challenge is to make these malicious updates appear benign to the central server's defense mechanisms. The 3DFed^[32] framework addresses this through a multi-faceted, adaptive approach: backdoor training with constraints, noise masks, and a decoy model. An overview of the 3DFed framework is shown in Fig. 4.

The attacker's local training objective is augmented with a constraint term to evade defenses like norm clipping. The loss function for crafting the local backdoor model $\omega_{adv,t}$ at epoch t is defined as:

$$\mathcal{L} = (1 - \beta) \cdot \mathcal{L}_{task} + \beta \cdot \|\omega_{adv} - G_t\|_2, \quad (10)$$

where $\mathcal{L}_{task}$ is the standard task loss combined with the backdoor objective, G_t is the current global model, and β controls the strength of the constraint. This term minimizes the Euclidean distance between the malicious model and the global model, producing a stealthier update with a smaller norm.

To further evade detection based on anomaly in neuron activations or update similarities, 3DFed applies optimized noise masks Δ_i to the backdoor model, creating a set of noisy models $X_i = \omega_{adv} + \Delta_i$. The noise is optimized using the following composite loss function:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{UPs} + (1 - \alpha) \cdot \mathcal{L}_{norm} + \lambda \cdot \mathcal{L}_{constrain}. \quad (11)$$

Here, $\mathcal{L}_{UPs}$ increases the Update Energies of selected neurons to mimic benign distributions, $\mathcal{L}_{norm}$ controls the magnitude of the noise masks, and $\mathcal{L}_{constrain} = \|\sum_{i=1}^m \Delta_i\|_2$ ensures the sum of all masks is near zero, preserving the backdoor effect upon aggregation. This disperses the malicious updates, reducing their pairwise cosine similarity and evading defenses like FoolsGold.

A cornerstone of 3DFed is its *indicator mechanism* for black-box adaptation. Attackers identify redundant neuron parameters with low curvature (second derivative of the loss) and implant indicators by scaling their updates by a large factor κ . In the subsequent epoch, they read the feedback by comparing the global model's change on these parameters to their own submitted change. The status of the previous attack (e.g., accepted or rejected) is inferred from the ratio:

$$feedback_i = \frac{\theta_{G_t - G_{t-1}, I_i}}{\theta'_{\omega_i - G_{t-1}, I_i}} = \frac{\theta_{G_t - G_{t-1}, I_i}}{\kappa \cdot \theta_{\omega_i - G_{t-1}, I_i}}. \quad (12)$$

If $feedback_i \leq 1/\kappa$, it strongly indicates the model was rejected, as the global change is dominated by other benign updates. This feedback loop allows dynamic adjustment of hyperparameters like the noise intensity α without any white-box knowledge of the defense.

3.3 Comparative analysis of poisoning attacks

The representative attacks summarized in Table 3 show that poisoning attacks in federated learning differ not only in their technical mechanisms but also in their practical threat profiles. In particular, their impact depends on the attacker's objective, required capability, knowledge of the FL system, scalability, stealthiness, and adaptivity. Therefore, a comparative analysis across these dimensions can help clarify the conditions under which each type of attack becomes more threatening.

From the perspective of effectiveness, different attacks optimize different forms of damage. Optimization-based poisoning attacks, such as OPT, directly target the mathematical properties of aggregation rules and aim to maximize the deviation of the global model. Therefore, they are often effective when the attacker has sufficient knowledge of the target aggregation mechanism. Fake-client attacks, such as MPAF, improve their attack strength by increasing the number or influence of malicious participants, thereby pulling the global model toward an attacker-selected base model. In contrast, contribution-evaluation poisoning attacks, such as ACE, may not immediately collapse the global model accuracy, but they can effectively manipulate contribution scores and undermine the fairness of participant evaluation.

These attacks also differ in their required adversarial capability and knowledge. Optimization-based attacks usually require the attacker to understand or approximate the aggregation rule, making them more dependent on defense-specific knowledge. Fake-client attacks require the ability to create or control a large number of virtual clients, so their feasibility is closely related to client admission control and identity management in practical FL systems. Contribution-evaluation poisoning attacks rely more on historical model information and the contribution-evaluation workflow. Backdoor-oriented attacks further vary in their assumptions: ProFlip requires access to the deployed model or runtime storage, whereas adaptive backdoor attacks such as 3DFed rely on malicious clients that can adjust their behavior according to feedback from the training process.

Scalability is another important distinction. Fake-client attacks are naturally scalable because the attacker can amplify the poisoning effect by increasing the number of forged participants, although this advantage may be weakened by strong authentication or reputation mechanisms. Contribution-evaluation attacks can also be scalable when they exploit general weaknesses in the evaluation workflow rather than a specific aggregation rule. By contrast, optimization-based attacks may be less scalable across different FL systems because their effectiveness often depends on the target aggregation rule. Deployment-stage attacks, such as bit-flip backdoors, are powerful but less

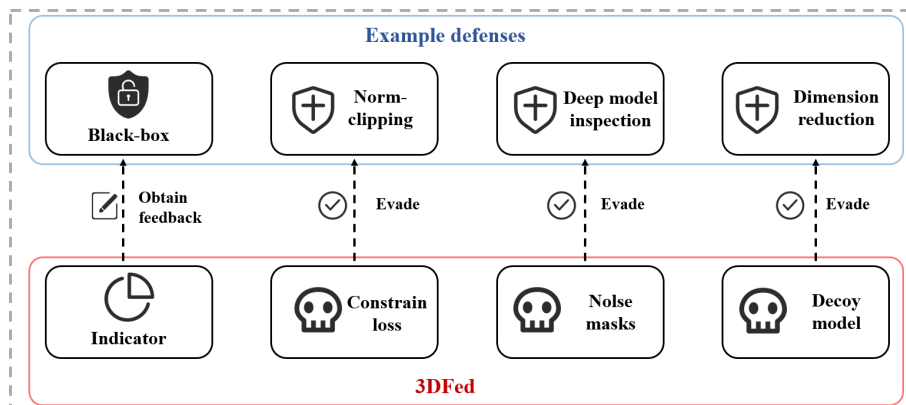


Fig. 4 Overview of 3DFed framework.

scalable in typical FL training scenarios because they require post-training access to model parameters or memory.

Stealthiness and adaptivity are closely related but not identical. Stealthiness refers to whether malicious behavior can avoid being detected by the server or defense mechanism, whereas adaptivity refers to whether the attacker can adjust the attack strategy according to the defense process. ACE and ProFlip are highly stealthy because they either disguise malicious updates as high-contribution updates or implant backdoors after deployment. However, adaptive attacks such as 3DFed go further by jointly optimizing backdoor effectiveness and defense evasion, for example through multi-objective optimization, noise masks, or feedback-based adjustment. This indicates that future poisoning attacks may increasingly shift from direct perturbation to defense-aware and feedback-driven strategies.

Overall, poisoning attacks should not be compared only by whether they reduce accuracy or achieve a high attack success rate. A more complete understanding should jointly consider the attack objective, required capability, knowledge assumption, scalability, stealthiness, and adaptivity. This perspective also explains why a defense that is effective against one attack type may not generalize to another: different attacks exploit different weaknesses in the FL pipeline, including aggregation rules, contribution evaluation, client population management, deployment security, and feedback-driven defense evasion.

4 Defense strategies of poisoning attacks

Anomaly detection-based defenses are built on the idea that malicious updates leave observable traces that differ from benign training behavior. These traces may appear as abnormal distances, directions, norms, activation patterns, feature distributions, or update trajectories. Therefore, the central problem is how to define a reliable boundary between normal client heterogeneity and malicious deviation, especially when the data are non-IID or the attacker is adaptive. Tables 4 and 5 summarize the major defense paradigms against poisoning

Table 4. Comparison of defense mechanisms against poisoning attacks in federated learning.

Category	Defense strategy	Attack type	Key limitation
Robust aggregation	Median, trimmed mean	Byzantine	Ineffective against backdoors
Update filtering	Similarity/anomaly detection	Byzantine	Adaptive attacks
Backdoor detection	Trigger inference	Backdoor	Trigger dependency
Backdoor mitigation	Fine-tuning, unlearning	Backdoor	Extra computation
Hybrid defense	Joint filtering and mitigation	Both	Increased complexity

Table 5. Comparison of defense strategies in FL.

Defense	Mechanism	Limitation
RA	Robust aggregation (median, trimmed mean)	Ineffective against backdoors
UF	Update filtering (similarity/anomaly detection)	Vulnerable to adaptive attacks
BD	Backdoor detection (trigger inference)	Depends on trigger knowledge
BM	Backdoor mitigation (fine-tuning, unlearning)	Extra computation cost
HD	Hybrid defense (joint filtering + mitigation)	Increased complexity

RA: robust aggregation; UF: update filtering; BD: backdoor detection; BM: backdoor mitigation; HD: hybrid defense.

attacks in federated learning. However, the limitations of these defenses should not be interpreted merely as isolated empirical weaknesses. Instead, they often arise from specific assumptions about the distribution, observability, and separability of client updates. In particular, the effectiveness of many defenses depends on whether benign and malicious updates can be reliably distinguished by distance, direction, norm, activation patterns, or validation behavior. These assumptions become fragile under adaptive attacks, where adversaries deliberately optimize malicious updates to resemble benign ones, and under non-IID settings, where benign updates naturally become more diverse. Therefore, we further analyze the failure mechanisms of existing defense strategies from the perspective of their underlying assumptions and implementation constraints.

4.1 Analytical discussion on defense limitations

The limitations of existing defense strategies are closely related to their underlying assumptions about benign and malicious client updates. Robust aggregation methods, such as Krum, median, and trimmed mean, usually assume that benign updates occupy a dominant and statistically stable region in the update space. Distance-based rules further rely on the geometric separability between benign and malicious updates, while coordinate-wise rules assume that malicious deviations can be suppressed independently in each dimension. These assumptions are reasonable against simple Byzantine perturbations, but they become fragile when attackers are aware of the defense mechanism. A rule-aware adversary may construct malicious updates that stay close to the benign cluster, or keep each coordinate within an acceptable statistical range while still accumulating a biased global direction. In this case, the degradation of the defense is mainly caused by a mismatch between its robustness assumption and the attacker's adaptive strategy, rather than by a simple implementation flaw.

Non-IID data further complicates this separation. In heterogeneous federated learning, benign clients may naturally generate updates with diverse directions, magnitudes, and class-specific gradients due to local data skew. Consequently, the boundary between benign heterogeneity and malicious deviation becomes ambiguous. If a defense regards large distance, low similarity, or abnormal norm as direct evidence of maliciousness, it may mistakenly filter benign clients whose data distributions differ from the majority. However, if the detection threshold is relaxed to tolerate such heterogeneity, carefully crafted malicious updates may also be accepted. Therefore, non-IID settings do not merely weaken empirical robustness; they directly challenge the separability assumption that many filtering and aggregation methods depend on.

Beyond these assumption-level limitations, implementation choices can also affect the reliability of defense mechanisms. Many anomaly detection and update filtering methods depend on fixed thresholds, pre-defined similarity metrics, or validation data that may not represent the true distribution of participating clients. Such designs may perform well in controlled experiments, but they are less reliable in dynamic FL environments where client participation, data distributions, and attack strategies change over time. Backdoor detection methods face a similar issue because they often rely on recoverable trigger patterns, abnormal activation signatures, or specific inspection layers. Adaptive backdoor attacks can reduce these observable traces through distributed triggers, multi-objective optimization, or noise masks, making detection outcomes sensitive to the selected validation samples, model layers, and decision thresholds.

Privacy-preserving mechanisms introduce an additional trade-off. Differential privacy, secure aggregation, and homomorphic encryption can reduce information leakage, but they also restrict the server's

access to plaintext update statistics that are often used for anomaly detection. This creates a privacy–robustness tension: exposing more update information may improve detection, whereas stronger privacy protection may remove the signals needed to distinguish malicious updates from benign ones. Therefore, future defense designs should clarify not only whether a method is effective against a specific attack, but also which assumptions it relies on, how these assumptions behave under non-IID data, and whether its limitations are caused by fundamental assumptions, implementation choices, or privacy constraints.

4.2 Anomaly detection-based defense

Anomaly detection-based defense (ADD) is one of the core paradigms for addressing unknown malicious threats in FL[33]. Its core logic stems from the "statistical distinguishability between normal and abnormal behaviors"[34]. By establishing a cognitive boundary for the distribution characteristics of legitimate samples and the normal response patterns of models, it identifies malicious samples or abnormal model states that deviate from this boundary as threats. From the perspective of technical logic, anomaly detection-based defense can be further divided into two core branches: one focuses on "data-level anomalies", which are defense methods based on abnormal feature distribution identification; the other focuses on "model-level anomalies", which are backdoor attack defense methods based on deep model inspection.

4.2.1 Defense based on the identification of anomalous feature distribution

GradCAM is used to generate heat maps for each uploaded local model, converting the model into a low-dimensional visual representation. This enables the identification of malicious model updates by detecting anomalous GradCAM heat maps. Zheng et al. proposed a refined defense mechanism called GCAMA[35], which enhances the accuracy of anomalous feature distribution identification by combining GradCAM with an autoencoder (AE). Compared to traditional Euclidean distance-based methods, GCAMA focuses on gradient-

weighted activation feature distribution and uses AE to model and validate this distribution, achieving stronger detection performance. Its technical framework is shown in Fig. 5, with key steps for anomalous feature distribution identification summarized below: (1) Device-side feature extraction foundation; (2) Server-side GradCAM for feature distribution visualization; and (3) Quantitative detection of anomalous feature distribution.

Nguyen et al.[23] proposed FLAME, a federated learning defense framework that defends against backdoors by focusing on identifying the anomalous feature distribution of model updates, with its core logic being detecting deviations from the "normal feature distribution baseline" of benign model updates to identify and mitigate malicious behaviors. In the model clustering module, FLAME uses pairwise cosine distances to measure angular differences between model updates—benign updates have a concentrated angular feature distribution, while malicious ones deviate significantly[36]. It then uses the HDBSCAN algorithm to group benign updates with a normal distribution and label outlier updates with an anomalous angular distribution for elimination. In the weight clipping module, it takes the median of L2-norms of all updates as the clipping bound. Benign updates have a stable L2-norm distribution, while malicious ones often have abnormally large L2-norms to amplify attacks; clipping scales down updates with an anomalous L2-norm distribution to limit their impact on aggregation. In the noise injection module, it estimates model update sensitivity based on the normal distribution of benign model differences to determine the noise amount, injecting Gaussian noise into the global model to eliminate backdoor contributions while preserving benign performance.

Zhang et al. proposed FLDetector[37], a malicious-client detection method centered on the identification of anomalous feature distribution of model updates; its core logic is to distinguish benign and malicious clients through the difference in the consistency feature distribution of model updates. Benign clients generate model updates following FL algorithms and local data, so their updates adhere to the consistency of feature distribution as defined by the Cauchy mean value theorem, with the core formula:

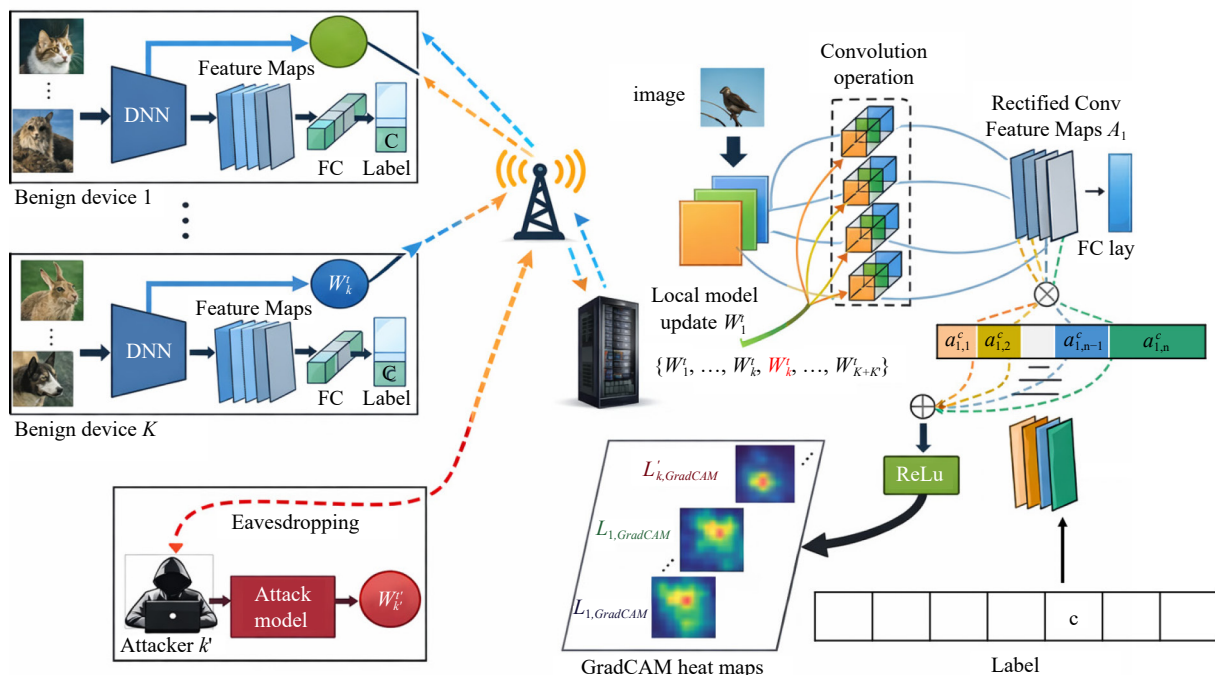


Fig. 5 System model of GCAMA.

$$g_i^t = g_i^{t-1} + \mathbf{H}_i^t \cdot (w_t - w_{t-1}) \quad (13)$$

where g_i^t and g_i^{t-1} are client i 's updates in the t -th and $(t-1)$ -th iterations, $\mathbf{H}_i^t$ is the integrated Hessian of client i , and w_t, w_{t-1} are global models of the two iterations. This formula encodes the stable consistency feature distribution of benign updates. In contrast, malicious clients craft updates arbitrarily, breaking this consistency and leading to an anomalous feature distribution of their updates. The server leverages the Cauchy mean value theorem to predict each client's update based on historical update data, uses Euclidean distance to measure the similarity between predicted and received updates, and calculates a suspicious score for each client. Finally, FLDetector applies k-means with Gap statistics to cluster the suspicious scores: since benign clients have concentrated suspicious score distributions and malicious ones have scattered, high-score distributions, it classifies clients in the cluster with larger average suspicious scores as malicious. After removing most malicious clients, Byzantine-robust FL methods can train accurate global models, with all processes revolving around the core of identifying anomalous feature distributions of model updates.

Ehsanul et al. proposed FLShield^[38], a federated learning defense framework that defends against poisoning attacks by centering on the identification of anomalous feature distributions of model updates, solving the validation subject and validation integrity dilemmas without relying on the server's clean datasets. It generates two types of representative models to capture normal feature distributions of benign updates and identifies anomalous ones: for bijective representative models, it takes each local update ω as the base, incorporates contributions of other updates via ReLU-weighted cosine similarity $s(\omega, u)$, and weights them by update magnitude ratio $|\omega|/|u|$, with the core:

$$\mathcal{E} = G_t + (1 - \tau)\omega + \tau \frac{\sum_{u \in \mathcal{U} - \{\omega\}} \left(s(\omega, u) \times u \times \frac{|\omega|}{|u|} \right)}{\sum_{u \in \mathcal{U} - \{\omega\}} \left(s(\omega, u) \times \frac{|\omega|}{|u|} \right)} \quad (14)$$

For cluster representative models, it uses K-Means to cluster local updates, averaging updates in each cluster to generate representative models, while malicious ones are scattered or form small abnormal clusters. In its defense pipeline, the server generates representative models, filters outlier evaluation results, selects the top 50% representative models, applies norm-bounded clipping to selected local updates, and aggregates to obtain the global model.

Hossein et al. present FreqFed^[39], and its core logic lies in transforming model weights (updates) into the frequency domain via the Discrete Cosine Transform (DCT) to distinguish between normal and anomalous feature distributions: benign updates retain core information in low-frequency components, showing a stable and concentrated low-frequency feature distribution, while malicious updates disrupt

this regularity, leading to an anomalous low-frequency feature distribution. FreqFed then extracts low-frequency component vectors from DCT results and uses the HDBSCAN density-based clustering algorithm to identify distribution anomalies. Benign vectors with normal low-frequency feature distribution cluster into a large main cluster, while malicious vectors with anomalous distribution are either scattered as outliers or form small abnormal clusters. The high-level overview of the framework FreqFed is shown in Fig. 6. FreqFed comprises three critical components for: (1) frequency analysis of local model updates; (2) model filtering; and (3) model aggregation. In the following, we give a brief explanation of the functionality of each component.

In general, all these methods take "feature distribution deviation" as the core criterion for judging malicious behaviors. Additionally, these methods differ in specific focuses: Activation mapping zeroes in on activation layer feature distributions to find abnormal activations linked to backdoors; FLAME analyzes model update feature distributions in federated learning for defense; FLDetector detects abnormal distributions in client model updates to differentiate benign and malicious ones; FLShield uses abnormal feature distribution identification to build defenses for federated learning systems; and FreqFed ensures system security during federated collaborative training by identifying anomalies in model feature distributions.

4.2.2 Backdoor attack defense based on deep model inspection

The core of backdoor attack defense methods based on deep model inspection is to break through the limitations of surface-level analysis. By conducting systematic inspections on deep-dimensional features of the model—such as internal structure, parameter correlation, and intermediate-layer representations—these methods excavate hidden traces left by backdoors.

Rieger et al. propose DeepSight^[40], a model filtering framework for mitigating FL backdoor attacks, with its core defense logic centered on the identification of anomalous feature distributions of model updates. In the filtering layer, it first extracts three feature metrics that characterize the training data distribution of model updates—benign updates, derived from normal local data, exhibit a concentrated and stable distribution of these features, while malicious updates show obvious deviations. It then clusters these features and calculates threshold exceedings from NEUPs to label updates: updates with feature distributions outside the normal cluster range or exceeding NEUP thresholds are marked as suspicious. The structure of the filtering layer is illustrated in Fig. 7. In the clipping layer, it targets malicious updates that bypass the first layer by using high scaling factors, with the core clipping formula: $\lambda_{t,i}^c = \min\left(1, \frac{S}{\|W_{t,i} - G_t\|}\right)$ where $W_{t,i}$ is the local

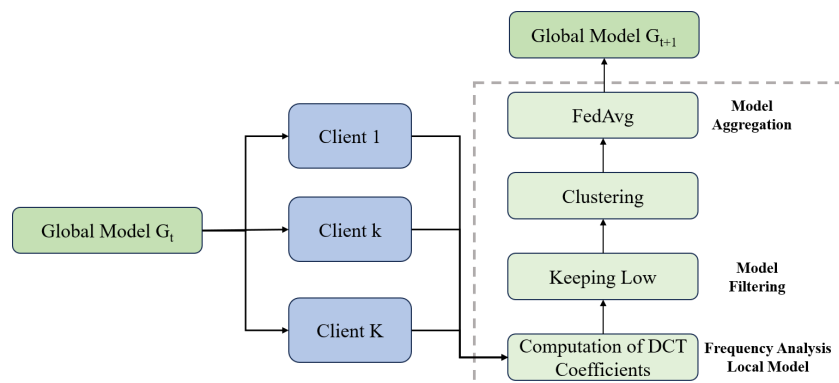


Fig. 6 System overview of FreqFed.

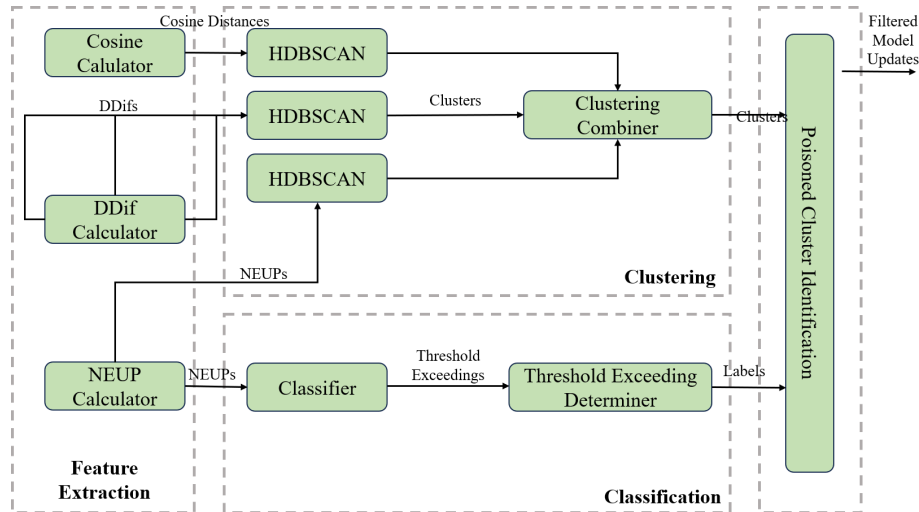


Fig. 7 Filtering layer of DeepSight.

model, G_t is the global model, and S is dynamically set as the median of L2-norms of all updates. Since benign updates have a stable L2-norm feature distribution, malicious updates with abnormally large L2-norms are scaled down to limit their impact. In the aggregation layer, it aggregates remaining clipped updates via FedAvg, and in the final round, aggregates by clusters, ensuring exclusion of updates with anomalous feature distribution. The structure overview is shown in Fig. 8.

Both the defense methods based on abnormal feature distribution identification and those based on deep model inspection take "capturing the feature differences between malicious models and benign models" as their core, but their technical paths differ significantly. The former focuses on the distribution deviation of surface-level features such as model parameters and activations, while the latter delves into the internal structure of models.

4.3 Defense based on privacy-preserving computing technology

4.3.1 Defense methods based on zero-knowledge proof technology

FL has an iterative, decentralized model training process. FL is vulnerable to both robust and privacy threats. Robustness threats include backdoor poisoning attacks and privacy threats^[41], and include membership and property inference attacks. Previous protections could

only deal with one type of threat, thus a hodgepodge of defenses. Differential Privacy (DP) provides a way to limit information leakage, and its two types of differential privacy, Local DP (LDP) and Central DP (DP), can be adapted to provide a unified defense against FL.

The core principle is to bound the influence of any single client's data or any single data record within a client's dataset on the final model by carefully injecting noise during the training process. This inherently limits the ability of malicious clients to implant an effective backdoor (a robustness gain) and obstructs an adversary's attempt to infer membership or properties of the training data.

In LDP, each client perturbs their model updates locally before uploading them to the server. It gives a good, minimally trusting privacy guarantee because the server really only ever sees noisy updates. The LDP mechanism M satisfies (ϵ, δ) -LDP if for all pairs of local datasets D, D' differing in one record and all possible outputs S :

$$P[M(D) \in S] \leq e^\epsilon P[M(D') \in S] + \delta. \quad (15)$$

In this definition, M denotes a randomized privacy mechanism, and D and D' denote two neighboring local datasets that differ in one data record. The set S represents any possible output event of the mechanism. The parameter ϵ controls the multiplicative privacy loss, while δ denotes the probability that the privacy guarantee may be violated beyond this bound. In general, a smaller ϵ corresponds to stronger privacy protection, but it may also require more noise and thus reduce model utility. In practice, this is implemented by having clients use DP-stochastic gradient descent (DP-SGD). Their local update Δ_k^r in

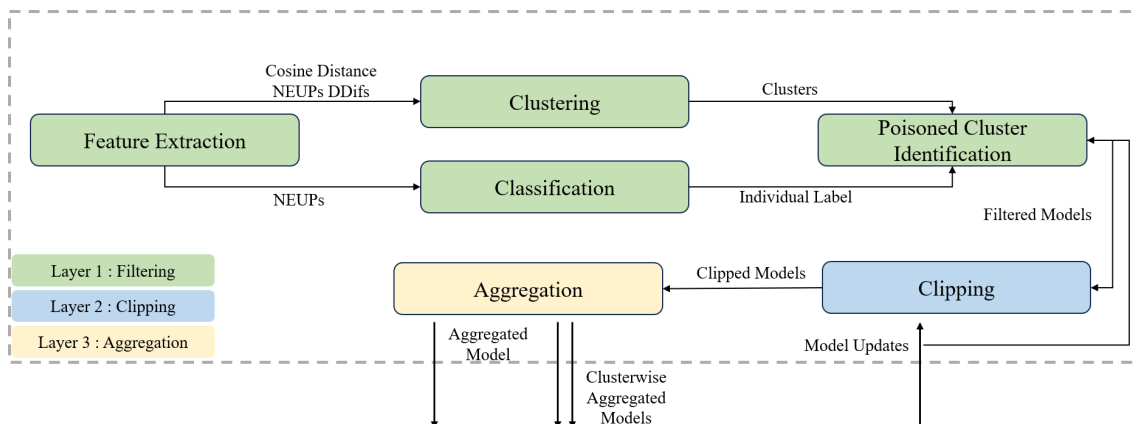


Fig. 8 Structure of DeepSight.

round r is computed under DP guarantees, typically by clipping the gradient's l_2 -norm to a bound S and adding Gaussian noise scaled to S and the sampling probability q :

$$\Delta_k^r \leftarrow \text{Clip}(\nabla\theta_k^r, S) + \mathcal{N}(0, \sigma^2 S^2 I). \quad (16)$$

In this DP-SGD update, Δ_k^r denotes the perturbed local update of client k in round r , and $\nabla\theta_k^r$ denotes the local gradient before privacy perturbation. The clipping operation $\text{Clip}(\cdot, S)$ bounds the l_2 -norm of the gradient by S , so that the sensitivity of each uploaded update is controlled. The Gaussian noise term $\mathcal{N}(0, \sigma^2 S^2 I)$ is then added to the clipped gradient, where σ determines the noise magnitude. Here, S refers to the clipping bound, which is different from the output event set S used in the DP definition above. The moments accountant technique is used to track the cumulative privacy budget (ϵ, δ) across training rounds.

In **Central DP (CDP)**, clients are trusted to send their true updates, and the server adds noise during the aggregation phase to provide *client-level* privacy. This ensures that the aggregated model does not reveal whether any particular client participated in the training. The server clips each client's update Δ_k^r to a maximum norm S and aggregates them before adding Gaussian noise:

$$\theta_{r+1} = \theta_r + \frac{1}{K} \sum_{k=1}^K \text{Clip}(\Delta_k^r, S) + \mathcal{N}(0, \sigma^2 S^2 I). \quad (17)$$

In this central DP aggregation formula, θ^r and θ^{r+1} denote the global models before and after aggregation in round r , respectively. K is the number of participating clients, and Δ_k^r is the update submitted by client k . The server first clips each client update using the bound S , aggregates the clipped updates, and then adds Gaussian noise with scale controlled by σS . Compared with LDP, CDP assumes that the server can access unclipped client updates before adding noise, and therefore it usually provides client-level privacy with a different trust assumption. The noise scale σ is determined by the target (ϵ, δ) values and the number of rounds, again managed via the moments accountant.

This dual approach provides a unified defense:

- **Against backdoor attacks:** The noise addition during LDP or CDP disrupts the carefully crafted malicious updates needed to embed a backdoor. For instance, on the EMNIST dataset with 2,400 clients and one attacker, CDP ($\epsilon = 3$) can reduce backdoor accuracy from 88% to 6%, while LDP ($\epsilon = 3$) reduces it to 10%, outperforming non-private defenses like norm bounding.

- **Against membership inference attacks:** The noise obfuscates the model's parameters, making it difficult for an adversary to distinguish whether a specific data point was part of the training set. Experiments show LDP ($\epsilon = 8.6$) and CDP ($\epsilon = 5.8$) can reduce the accuracy of a white-box membership inference attack from 75% to 55% and 52%, respectively, on CIFAR100.

- **Limitations against property inference:** While CDP can theoretically defend against property inference, it often requires a very small ϵ that severely degrades utility. LDP, providing record-level privacy, is generally ineffective against property inference attacks that target dataset-level characteristics.

4.3.2 Defense based on homomorphic encryption technology

In FL, decentralized training faces two major threats: (1) utility-side evasion attacks; and (2) privacy-side model inversion attacks (MIA). Prior defenses often target only one, leaving gaps. Federated Cryptography Defense (FCD) provides a unified solution by conducting training and inference entirely in an encrypted space via homomorphic encryption, simultaneously disrupting adversarial perturbations and blocking MIAs.

The core idea is to operate in an encrypted domain using a row-based transposition cipher with a shared secret key $\mathcal{K}$. This ensures perturbations in the original space lose effectiveness and reconstructed images from MIAs remain meaningless.

In **FCD Encryption**, each client transforms its data $\mathcal{X}_k$ by transposition and row-wise shuffling with key $\mathcal{K}$:

$$\mathcal{E}(\mathcal{X}_k) = \mathcal{R}_{\mathcal{K}}(\mathcal{X}_k^T) \quad (18)$$

where $\mathcal{R}_{\mathcal{K}}(\cdot)$ is row-wise shuffling, $\mathcal{K} = [\kappa_0, \kappa_1, \dots, \kappa_{h-1}]$, and h is image height.

To enable learning, FCD introduces a **distillation loss** combining cross-entropy on encrypted data and KL divergence between teacher predictions on clean data and student predictions on encrypted data:

$$\mathcal{L}_k = \mathcal{L}_{CE_k} + \alpha_k \mathcal{L}_{KLD_k} \quad (19)$$

$$\mathcal{L}_{KLD_k} = \sum_{i=1}^{N_k} \sum_{r=1}^R Q_{i,k}^r \log \frac{Q_{i,k}^r}{\mathcal{P}_{i,k}^r} \quad (20)$$

where $\mathcal{P}_{i,k}$ and $Q_{i,k}$ are predictions on normal and encrypted data, and α_k is a balancing weight.

The server also evaluates in encrypted space:

$$\mathcal{E}(\tilde{\mathcal{D}}_{\text{test}}) = \mathcal{R}_{\mathcal{K}}(\tilde{\mathcal{D}}_{\text{test}}^T) \quad (21)$$

ensuring poisoned data are transformed before testing.

FCD guarantees robustness through ζ -separability:

$$\|\mathcal{E}(\mathcal{X}_i) - \tilde{\mathcal{X}}_j\| \geq \zeta \quad (22)$$

so adversarial examples cannot remain close to valid encrypted data.

- **Evasion attacks (TM1):** Encryption disrupts perturbations. On GTSRB/KBTS, FCD maintains utility loss 2–6 units lower than FAT and RS at $A_p = 30\%$.

- **Model inversion (TM2):** MIA reconstructions remain encrypted. On CIFAR100, FCD reaches MSE 73.23, far above Laplacian (0.011) and ResSFL (0.05).

- **Theory:** Provides convergence guarantees, robustness to $\mu\rho$ -bounded evasion and ϵ -distorted MIA attacks.

The key advantages of FCD are: 1) Low visibility (unique gradients for each key); 2) low budget (linear time complexity $O(nh)$); 3) Multi-purpose efficiency. This framework gives a useful way to make FL more resistant and private from a defender's viewpoint, which is like handling lots of different threats with just one crypto change.

FL's decentralized and iterative training makes it vulnerable to both privacy inference attacks and security threats such as model poisoning. Most defenses focus on one dimension only, leading to fragmented protection. The Dual Defense Federated Learning (DDFed) framework integrates Fully Homomorphic Encryption (FHE) with a two-stage anomaly detection mechanism on encrypted updates, providing simultaneous defense against privacy and poisoning threats.

The key idea is secure aggregation via FHE, combined with anomaly detection over encrypted model updates using secure similarity computation and feedback-based collaborative filtering. Thus, the server never sees plaintext updates, while poisoned models can still be detected and filtered.

In **FHE-based secure aggregation**, each client C_i encrypts its local update:

$$W_i^{(t)} = \text{FHE_EncPK}(W_i^{(t)}), \quad (23)$$

and the server aggregates without decryption:

$$W_G^{(t)} = \sum_{i \in S^{(t)}} f_i^{(t)} \cdot W_i^{(t)}, \quad (24)$$

where $f_i^{(t)}$ are weights proportional to dataset sizes.

For **Secure Anomaly Detection**, DDFed uses two phases. In secure similarity computation, cosine similarity between encrypted updates and the previous global model is computed, with updates pre-normalized to avoid division. Gaussian noise $\Delta^{(t)}$ is added for (ϵ, δ) -DP:

$$s^{(t)} = \left\langle \frac{w_i^{(t)}}{\|w_i^{(t)}\|_2} + \Delta^{(t)}, \frac{w_G^{(t-1)}}{\|w_G^{(t-1)}\|_2} \right\rangle. \quad (25)$$

Then, in feedback-driven collaborative selection, clients decrypt similarity scores, filter by threshold, and return results. The server applies majority voting to decide the final benign set $S^{(t)}$.

The framework offers:

- **Privacy protection:** FHE ensures the server never accesses plaintext updates; DP noise prevents leakage via similarity scores.
- **Poisoning defense:** Malicious updates (e.g., IPM, SCALING, ALIE) are filtered out. On MNIST and FMNIST, DDFed sustains > 80% accuracy under 30% malicious clients, outperforming Krum and Median.
- **Efficiency:** Despite the overhead of FHE, the training adds only ~2s ($\approx 20\%$) per round, remaining practical for both cross-silo and cross-device FL.

Overall, DDFed provides cryptographic privacy, robust anomaly detection, and practical efficiency without altering the FL topology or requiring extra servers. It demonstrates that unified protection against both privacy and poisoning attacks is feasible by integrating FHE with secure computation.

4.4 Defenses based on client protection

4.4.1 Robustness enhancement based on client-side personalized learning

A critical dilemma in FL is that statistical heterogeneity amplifies vulnerability to poisoning attacks—malicious clients exploit decentralized data to corrupt global models, while traditional robust defenses often filter informative local updates and worsen fairness. Client-side personalized learning resolves this by enabling devices to learn tailored local models that balance global alignment and local adaptation. Li et al. focused on Ditto^[42], a global-regularized federated multi-task framework, to elaborate on its robustness mechanism against poisoning attacks, with rigorous mathematical modeling and empirical validation.

Ditto abandons the single global model paradigm, formulating a bi-level optimization problem to jointly learn personalized local models ($\{v_k\}_{k \in [K]}$, one per device k) and a shared global model (w^*). The regularization term constrains local models to align with the global model, retaining generalization while resisting poisoning.

For each device $k \in [K]$ (K : total clients), Ditto minimizes a locally regularized objective:

$$\begin{cases} \min_{v_k} h_k(v_k; w^*) := F_k(v_k) + \frac{\lambda}{2} \|v_k - w^*\|_2^2 \\ \text{s.t. } w^* \in \arg \min_w G(F_1(w), \dots, F_K(w)) \end{cases} \quad (26)$$

Ditto uses alternating optimization (Li et al., 2021, Algorithm 1): in round t , the server updates w^t via aggregation, then each selected client k updates v_k via SGD: $v_k^{t+1} = v_k^t - \eta \cdot (\nabla F_k(v_k^t) + \lambda \cdot (v_k^t - w^t))$. Here, η is the local learning rate, and $\lambda \cdot (v_k^t - w^t)$ pulls v_k toward w^t , avoiding excessive deviation while resisting poisoned global updates. Ditto's robustness stems from decoupling local/global models (reducing reliance on corrupted w^*) and optimizing λ for attack resistance.

In summary, client-side personalization enhances FL robustness by reducing reliance on poisoned global models. Ditto's bi-level objective and optimal λ^* balance local resistance and global generalization, outperforming robust baselines by 6% in average accuracy.

4.4.2 Defense based on client update integrity verification

Secure aggregation (SecAgg) ensures the privacy of client local updates by masking individual contributions, but it cannot verify the integrity of these updates—malicious clients may inject poisoned updates to degrade the global model or implant backdoors. Defense methods based on client update integrity verification aim to resolve this contradiction: under the premise of preserving client data privacy, they leverage cryptographic primitives to check whether local updates comply with pre-defined constraints (norm bounds, valid data formats). These methods filter out malicious updates before aggregation, while avoiding leakage of private information.

ACORN^[43] proposes the first SecAgg protocol with practical input validation, supporting L_0 (k-hotness), L_2 , and L_∞ norm checks, while improving computational and communication efficiency via lattice cryptography and Bulletproofs. Its core innovation lies in two parts: RLWE-SecAgg and ACORN variants. ACORN generalizes the prior PRG-SecAgg and instantiates it with RLWE to reduce computational complexity. For a client i with local update x_i and secret key sk_i , the encoding/decoding of updates follows:

$$\begin{cases} \text{Encode}(sk_i, x_i) = y_i := A \cdot sk_i + T(e + f) + x_i \mod q \\ \text{Decode}(sk, y) = (y - A \cdot sk \mod q) \mod T \end{cases} \quad (27)$$

where A is a public matrix, e, f are discrete Gaussian errors, and T, q are plaintext/ciphertext moduli; this reduces client computation from $O(\ell \log n)$ (in prior PRG-SecAgg) to $O(\ell)$, matching the insecure baseline. For input validation, ACORN converts range proofs into inner-product proofs via Lagrange's four-square theorem—clients prove $x' \circ x' + u \circ u + v \circ v + w \circ w = c$ and $\|x' \|_2 \|u \|_2 \|v \|_2 \|w \|_2 < \sqrt{q}/4$, using Bulletproofs to shrink proof size to $O(\log \ell)$ and limit communication overhead to just 0.1X more than SecAgg without validation. ACORN further offers two variants: ACORN-detect detects malicious clients, while ACORN-robust uses neighbor commitment consistency checks to exclude them, though it requires $6 + O(\log n)$ rounds and only supports a semi-honest server.

Addressing the round complexity and limitations in server security of ACORN, Ma et al.^[44] designs a 3-round SecAgg with input validation, achieving malicious security against both clients and servers. It adopts a two-layer aggregation framework: the outer layer uses Regev encryption to encrypt client updates x_i as $y_i = A \cdot k_i + e_i + \Delta \cdot x_i$ (where k_i is the encryption key, $\Delta = \lfloor q/p \rfloor$), and the server aggregates $y = \sum y_i = \text{Enc}(\sum k_i, \sum x_i)$; the inner layer samples C helper clients, and each client secret-shares k_i via packed Shamir sharing ($s_i^{(j)} = \text{SS.share}(k_i, [C], d)$)—helpers aggregate shares to let the server reconstruct $k = \sum k_i$ for decryption. For input validation, Armadillo uses Nova (a recursive ZKP) to prove three core constraints: Shamir sharing consistency ($\langle w, s_i \rangle = 0$), key-share binding ($k_i = M \cdot s_i$), and encryption correctness ($y_i = A \cdot k_i + e_i + \Delta \cdot x_i$ with norm bounds on x_i and e_i). These constraints are merged into inner-product proofs, reducing client computation by 1.5X compared to ACORN-robust, and the 3-round design makes it suitable for large-scale, latency-sensitive FL scenarios.

Unlike ACORN and Armadillo, which are centered on fixed integrity constraints, EIFFeL^[45] proposes a general framework for secure aggregation of verified inputs (SAVI), supporting arbitrary integrity predicates expressed as arithmetic circuits. It leverages Secret-Shared Non-Interactive Proofs (SNIP) and Shamir's sharing for distributed verification: each client i generates a SNIP proof π_i for $\text{Valid}(u_i) = 1$, then splits u_i and π_i via Shamir sharing: $\{(u_{i,j}), \Psi_{u_i}\} \leftarrow \text{SS.share}(u_i, [n], m+1)$ and $\{(i, \pi_{i,j}), \Psi_{\pi_i}\} \leftarrow \text{SS.share}(\pi_i, [n], m+1)$. Clients decrypt and verify received shares, flagging invalid ones; the server marks clients with $\geq m+1$ flags as malicious. For proof verification, clients generate summaries $\sigma_{i,j} = ((j, w_{i,j}^{\text{out}}), (j, \lambda_{i,j}))$, and the

server uses Reed-Solomon robust reconstruction to get $w_i^{out} = \text{SS.robustRecon}(\{(j, w_{i,j}^{out})\})$ and $\lambda_i = \text{SS.robustRecon}(\{(j, \lambda_{i,j})\})$, accepting u_i only if $w_i^{out} = 1$ and $\lambda_i = 0$. Finally, the server aggregates valid updates via $U = \text{SS.robustRecon}(\{(i, U_i)\})$ ($U_i = \sum_{j \in C^*} u_{j,i}$, C^* is malicious clients).

Defense methods based on client update integrity verification have become a critical direction for FL security, as they fill the gap of SecAgg in integrity checks while preserving privacy. ACORN pioneers the integration of input validation into SecAgg via RLWE and Bulletproofs, laying the foundation for efficiency and validity balance. Armadillo optimizes round complexity to constant rounds, making it suitable for latency-sensitive FL scenarios. EIFFeL achieves the highest flexibility by supporting arbitrary integrity predicates, enabling customization for diverse attack scenarios.

4.5 Defense based on a multi-layered defense framework

Single-layer defenses often struggle to balance attack mitigation effectiveness, model utility preservation, and adaptability to diverse attack types. Defense methods based on multi-layered defense frameworks address this limitation by integrating complementary defense components into a cohesive pipeline. These components work sequentially or in parallel to cover the entire attack lifecycle: first detecting potential malicious model updates, then limiting the impact of undetected malicious updates, and finally eliminating residual adversarial influences—all while minimizing degradation of the global model's benign performance. Such frameworks avoid the "one-size-fits-all" flaw of single-layer defenses, making them more robust to complex FL attack scenarios^[46].

Targeting the dilemma where existing backdoor defenses either rely on specific attack models or degrade benign performance via excessive noise, FLAME^[23] proposes a three-layer defense framework that integrates dynamic clustering, adaptive clipping, and adaptive noise injection. The first layer, dynamic clustering, leverages HDBSCAN to identify malicious models with large angular deviations from benign ones. It measures the angular difference between local model updates using the cosine distance metric: $c_{ij} = 1 - \frac{W_i W_j}{\|W_i\| \|W_j\|}$. By setting the minimum cluster size to more than half of the total clients ($\frac{n}{2} + 1$), FLAME ensures the main cluster contains mostly benign models, while labeling models outside this cluster as outliers and filtering them out. The second layer, adaptive clipping, addresses malicious models scaled up to amplify attack impact: it computes the clipping bound S_t as the median of the L_2 -norms of all local model updates ($S_t = \text{Median}(e_1, e_2, \dots, e_n)$, where $e_i = \|W_i - G_{t-1}\|$ and G_{t-1} is the previous global model) and clips each admitted model's weight vector as:

$$W_{b_i}^c = G_{t-1} + (W_{b_i} - G_{t-1}) \cdot \text{MIN}(1, \frac{S_t}{e_{b_i}}) \quad (28)$$

The third layer, adaptive noise injection, eliminates residual backdoors using DP-inspired Gaussian noise. Based on a formally proven noise bound, the noise scale is determined as $\sigma_G = \lambda \cdot S_t$, where $\lambda = \frac{1}{\epsilon} \sqrt{2 \ln \frac{1.25}{\delta}}$ (ϵ and δ are DP parameters). This noise scale is derived from the clipping bound S_t , ensuring the noise is sufficient to neutralize backdoors while avoiding excessive benign performance loss—thanks to the prior two layers reducing the required noise amount.

Unlike FLAME, which focuses on balancing noise injection and benign performance, FLShield^[38] addresses two core dilemmas in FL

validation—"validation subject dilemma" (how to select validation models that are both privacy-preserving and representative) and "validation integrity dilemma" (how to ensure validation results from potentially malicious validators are reliable)—via a four-layer framework integrating representative model generation, class-level validation (LIPC), anomaly filtering, and adaptive clipping. To solve the validation subject dilemma, FLShield first generates "representative models" by fusing multiple local updates. Specifically, each local update ω acts as the base, and the "sibling contributions" of other local updates are weighted by similarity to compute the representative model $\mathcal{E}$, as shown in the formula:

$$\mathcal{E} = G_t + (1 - \tau)\omega + \tau \frac{\sum_{u \in \mathcal{U} - \{\omega\}} (s(\omega, u) \times u \times \frac{|\omega|}{|u|})}{\sum_{u \in \mathcal{U} - \{\omega\}} (s(\omega, u) \times \frac{|\omega|}{|u|})} \quad (29)$$

Here, $s(\omega, u)$ denotes ReLU-weighted cosine similarity, and τ regulates the ratio of sibling contributions. Such representative models resist gradient inversion attacks while maintaining generalization in validation.

To address the validation integrity dilemma, FLShield proposes the Loss Impact Per Class (LIPC) metric. This metric measures the class-wise loss difference between the representative model $\mathcal{E}$ and the previous global model G , ensuring consistent validation results among benign validators even in non-IID scenarios. The third layer filters anomalous validation results via elliptic envelope outlier detection, and the fourth layer applies norm-bounded clipping to selected local updates to block malicious updates from bypassing validation.

In summary, multi-layered defense frameworks represent a mature direction in FL poisoning defense, as they address the inherent limitations of single-layer methods by integrating complementary components. FLAME pioneers the integration of dynamic clustering, adaptive clipping, and theoretically grounded noise injection, providing a paradigm for balancing backdoor elimination and performance preservation. FLShield, meanwhile, innovates around validation mechanisms, solving core dilemmas in FL client-side validation and expanding the applicability of validation-based defenses to privacy-sensitive scenarios. Both frameworks demonstrate that multi-layered design—by covering detection, impact limitation, and residual threat elimination—can achieve robust defense against diverse poisoning attacks while maintaining the utility and privacy of FL systems.

4.6 Robust defense methods under non-IID data

In the non-IID scenario, the heterogeneity of client local data in class distribution not only exacerbates the challenge of model training but also provides a hidden space for poisoning attacks. Malicious updates may be mistaken for normal deviations caused by data heterogeneity, while benign edge updates may be misjudged as anomalies^[47]. Existing research has proposed robust defense methods that combine attack detection capabilities and data adaptability by integrating visualization analysis, cryptographic protocols, and other technologies. For example, the GCAMA framework proposed by Zheng et al. breaks through the limitations of traditional detection based on Euclidean distance. Malicious update recognition in non-IID scenarios is achieved through the collaboration of Gradient weighted Class Activation Mapping (GradCAM) and Autoencoder (AE). The core logic lies in transforming high-dimensional model parameters into interpretable visual features and removing the confusion between data heterogeneity and maliciously updated features. Specifically, the server randomly selects images covering all categories from the test set and inputs them into the local model update w_l uploaded by each client. The feature map A_l is generated through the last convolutional layer, and the

neuron importance weight is calculated based on the gradient of class activation scores:

$$\alpha_{l,m}^{(c)} = \frac{1}{B \bar{A} H} \sum_{i=1}^B \sum_{j=1}^H \frac{\partial Y^{(c)}}{\partial A_l^m(i, j)}, \forall l \in \mathcal{K} \cup \mathcal{M} \quad (30)$$

Subsequently, the ReLU activation function is used to generate a single channel GradCAM heatmap $L_{l, \text{textGradCAM}}^{(c)}$:

$$L_{l, \text{GradCAM}}^{(c)} = \text{ReLU} \left(\sum_{m=1}^n \alpha_{l,m}^{(c)} A_l^m \right) \quad (31)$$

This heatmap transforms the decision focus of model updates into a low-dimensional visual representation, and the autoencoder further reconstructs the heatmap. The reconstruction error R_l is defined by calculating the mean square error, and a threshold $\alpha_{l,m}^{(c)} = \frac{1}{B \bar{A} H} \sum_{i=1}^B \sum_{j=1}^H \frac{\partial Y^{(c)}}{\partial A_l^m(i, j)}$, $\forall l \in \mathcal{K} \cup \mathcal{M}$ is set based on statistical characteristics to determine malicious updates with errors exceeding the threshold. Under the non IID setting of CIFAR-10 and GTSRB datasets, GCAMA significantly improved the testing accuracy of ResNet-18 and MobileNetV3 Large models compared to Multi Krum and FAA-DL methods, demonstrating its strong adaptability to data heterogeneity.

The ELSA protocol^[48] proposed by Rathee et al. addresses the trade-off between gradient privacy and robustness in non-IID scenarios. It adopts a dual-server distributed trust architecture and achieves efficient defense through client-generated cryptographic association information. ELSA innovatively uses the client as a non trusted cryptographic association source, generates OT associations and Beaver triplets, and secretly shares them with dual servers. The servers verify the validity of the associations through sacrifice techniques, avoiding the high cost of generating associations through server interaction in traditional protocols. The core defense mechanism is ℓ_2^2 defense, which calculates the sum of squares of gradient vectors through Beaver triplets and combines component level bit length constraints to avoid finite-loop overflow, achieving filtering of boosted gradients:

$$\ell_2^2 = \sum_{i=1}^m x_i^2 \leq \mu^2, \quad |x_i| \leq 2^w - 1 \quad (32)$$

In the non-IID data scenario, ELSA achieves an eight-times runtime improvement compared to the Prio protocol and a 305-times efficiency optimization compared to RoFL through single-round client communication and lightweight server verification, while ensuring gradient privacy.

The EIFFeL^[45] system proposed by Chowdhury et al. solves the conflict between update integrity verification and privacy protection in non-IID scenarios through the fusion of secret sharing non-interactive proof (SNIP) and Shamir threshold secret sharing. EIFFeL treats clients as mutual validators, and each client updates u_i to generate SNIP proofs π_i . The updates and proofs are then split into secret shares and distributed to other clients using the Shamir threshold scheme (threshold $[m+1]$). The server uses the robust reconstruction feature of Reed–Solomon codes to recover proof digests $\sigma_{ji} = ((i, w_{ji}^{out}), (i, \lambda_{ji}))$ from multiple client shares, verifying $w_{out} = 1$ and $\lambda = 0$ to determine the validity of the update. This mechanism does not rely on the assumption of global data distribution, and avoids the problem of single verification standard failure caused by non-IID data through distributed verification between clients. The FreqFed proposed by Fereidooni et al. breaks through the feature confusion of non-IID data from a frequency domain perspective and achieves malicious update detection through discrete cosine transform (DCT) and density clustering. The core insight is that the energy of neural network weights is mainly concentrated in low-frequency components,

and malicious updates can introduce abnormal low-frequency patterns, which can be effectively separated from the high-frequency differences caused by non-IID data. FreqFed first performs DCT transformation on the local model weights W_i^t to obtain V_i^t , extracts low-frequency components F_i^t , and then constructs a distance matrix based on cosine distance:

$$\text{dist}_{ij} = 1 - \text{CosineSim}(F_i, F_j) \quad (33)$$

Identify the largest cluster (benign update) through HDBSCAN clustering, and only aggregate model updates within that cluster. On various non-IID datasets such as PROTEINS and Reddit, FreqFed was able to maintain main task accuracy similar to non-attack scenarios against attacks such as Label Flipping and Constrained, and Scale, and was not affected by data distribution heterogeneity, demonstrating its cross-scenario adaptation ability.

5 Future work

While current research has made significant progress in identifying and mitigating poisoning attacks in federated learning, numerous challenges remain in complex real-world applications. First, robustness in heterogeneous environments remains a fundamental open problem. Most existing defense mechanisms implicitly assume independent and identically distributed (IID) data, yet in practice, client data are often highly non-IID and exhibit significant statistical skew. Under such conditions, a key unresolved question is how to formally distinguish benign gradient deviations caused by data heterogeneity from malicious data perturbations introduced by adversaries. This raises the need for theoretically grounded criteria or identifiable boundaries that characterize the separability between normal and adversarial updates. At the methodological level, another critical challenge is how to design aggregation and detection mechanisms that are inherently aware of distributional bias, enabling accurate filtering of poisoned updates while avoiding excessive false positives that could degrade model convergence. Addressing these issues requires not only robust statistical modeling of client updates but also adaptive mechanisms capable of tolerating long-tail data distributions without sacrificing security.

Second, the tension between privacy preservation and security defense remains unresolved and calls for fundamentally new solutions. While techniques such as homomorphic encryption and secure multi-party computation are widely adopted to protect user data, they prevent the server from directly inspecting model updates, thereby rendering many existing effective poisoning detection methods ineffective. An important open question is how to perform reliable anomaly detection directly in the encrypted domain, for example by designing verifiable statistics, encrypted-distance measures, or zero-knowledge-based validation mechanisms that can certify the legitimacy of updates without revealing their content. Beyond this, there is a pressing need to rethink the adaptability and evaluation of defense systems. As poisoning attacks evolve toward more stealthy and adaptive strategies, future defenses must transition from static filtering to dynamic, game-theoretic, or learning-based approaches that can anticipate adversarial behavior. At the system level, another open challenge is the construction of a unified and standardized benchmarking framework that jointly considers diverse attack types, varying degrees of data heterogeneity, and multiple performance dimensions—including accuracy, robustness, privacy leakage, and computational overhead—so as to enable fair and reproducible comparison across different defense methods. Ultimately, these challenges point toward the necessity of developing decentralized and adaptive federated learning

paradigms that do not rely on a fully trusted central server and remain robust against continuously evolving adversaries.

Beyond attack-specific defense design, poisoning robustness should also be connected to the broader goal of trustworthy and intelligent distributed learning. In highly heterogeneous FL systems, the boundary between benign distributional shift and malicious deviation is often uncertain. Therefore, future defense mechanisms should not only make binary benign/malicious decisions, but also quantify the uncertainty of anomaly detection, client reliability estimation, and model prediction under potential attacks. Recent studies on uncertainty-aware FL provide useful inspiration for this direction. For example, FedSI introduces federated subnetwork inference to support efficient uncertainty quantification in personalized FL, showing that uncertainty modeling can be made scalable under heterogeneous client distributions^[49].

Adaptive and personalized modeling is another important direction. Since clients may differ in data distributions, computation resources, communication conditions, and risk levels, a single global decision boundary may be insufficient for distinguishing malicious updates from legitimate heterogeneous updates. Personalized FL, adaptive trust modeling, and Bayesian FL can provide client-specific robustness criteria, dynamic thresholds, and locally aware defense strategies. Along this line, FedKT explores a lightweight Bayesian federated learning framework for privacy-preserving consumer devices, suggesting that future FL systems should jointly consider uncertainty, privacy, personalization, and resource efficiency^[50]. These studies indicate that poisoning defense should be viewed not only as an isolated security module, but also as part of a broader trustworthy distributed intelligence framework.

Finally, scalable real-world deployment remains a critical challenge. Practical FL systems usually involve intermittent participation, resource-constrained devices, communication bottlenecks, and incomplete knowledge of client identities. Therefore, future poisoning defenses should be evaluated not only by clean accuracy or attack success rate, but also by scalability, communication cost, computational overhead, false-positive rate, privacy leakage, and compatibility with large-scale client management.

6 Conclusions

This paper systematically explores the security challenges of FL, with a focus on poisoning attacks, the primary threat to FL's decentralized and privacy-preserving attributes. We categorized poisoning attacks into untargeted Byzantine attacks and targeted backdoor attacks, analyzing their technical mechanisms, stealth strategies, and impact paths; we also summarized mainstream defense paradigms, including anomaly detection, privacy-preserving verification, client-side protection, and multi-layered frameworks, clarifying their effectiveness in addressing different attacks and non-IID scenarios. At the core of FL security is a dynamic adversarial balance: attackers enhance stealthiness, adaptability, and resource efficiency, while defenses evolve from superficial parameter analysis to deep semantic detection and from single-scenario specialization to multi-modal generality. Looking ahead, attacks will trend toward multi-modal integration and fine-grained targeting, while defenses focus on privacy-robustness fusion, non-IID adaptability, and proactive prevention.

To resolve unresolved FL security challenges—such as handling mixed attacks, adapting to edge device constraints, and balancing privacy, robustness, and efficiency—future research should prioritize key directions: developing cross-scenario universal defense frameworks, optimizing lightweight edge defenses, advancing formal robustness verification, achieving synergistic optimization of three core

goals, defending specialized FL paradigms, and promoting engineering practice and standardization. By comprehensively clarifying the adversarial landscape of poisoning attacks and defenses and outlining promising research paths, this paper aims to provide actionable references for FL security researchers and practitioners, accelerating the development of secure, robust, and practical FL systems.

Author contributions

The authors confirm contribution to the paper as follows: study conception and design: Shan X, Zhang M, Yu H; data collection: Shan X, Zhang J; analysis and interpretation of results: Shan X, Zhang J, Yang Z; draft manuscript preparation: Shan X, Zhang J, Zhang M. All authors reviewed the results and approved the final version of the manuscript.

Data availability

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (2022YFB3103104), the National Natural Science Foundation of China (62271456), Beijing Municipal Natural Science Foundation (4232018), Major Research Plan of National Natural Science Foundation of China (92167102), the R&D Program of Beijing Municipal Education Commission (KM202210005026), and the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (CIT&TCD20190308).

Conflict of interest

The authors declare that they have no conflict of interest.

Dates

Received 27 March 2026; Revised 25 May 2026; Accepted 16 June 2026; Published online 20 September 2026

References

- [1] Zhang C, Xie Y, Bai H, Yu B, Li W, et al. 2021. A survey on federated learning. *Knowledge-Based Systems* 216:106775
- [2] Rieke N, Hancox J, Li W, Milletari F, Roth HR, et al. 2020. The future of digital health with federated learning. *npj Digital Medicine* 3(1):1–7
- [3] Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, et al. 2020. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports* 10(1):12598
- [4] Albshaier L, Almarri S, Albuali A. 2025. Federated learning for cloud and edge security: a systematic review of challenges and ai opportunities. *Electronics* 14(5):1019
- [5] Geiping J, Bauermeister H, Dröge H, Moeller M. 2020. Inverting gradients – how easy is it to break privacy in federated learning? *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020, Vancouver, Canada. pp. 16937–16947 https://proceedings.neurips.cc/paper_files/paper/2020/file/c4ede56bbd98819ae6112b20ac6b145-Paper.pdf
- [6] Yin H, Mallya A, Vahdat A, Alvarez JM, Kautz J, et al. 2021. See through gradients: Image batch recovery via gradinversion. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 20–25 June 2021, Nashville, TN, USA. USA. IEEE. pp. 16332–16341 doi: [10.1109/cvpr46437.2021.01607](https://doi.org/10.1109/cvpr46437.2021.01607)

- [7] Xu W, Yang H, Zhang M, Pan X, Liu W, et al. 2022. Retinal vessel segmentation with vae reconstruction and multi-scale context extractor. *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI), Kolkata, India, 28–31 March 2022*. USA: IEEE. pp. 1–5 doi: [10.1109/ISBI52829.2022.9761563](https://doi.org/10.1109/ISBI52829.2022.9761563)
- [8] Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang YC, et al. 2020. Federated learning in mobile edge networks: a comprehensive survey. *IEEE Communications Surveys & Tutorials* 22(3):2031–2063
- [9] Adam M, Albaser A, Baroudi U, Abdallah M. 2025. Survey of multi-modal federated learning: Exploring data integration, challenges, and future directions. *IEEE Open Journal of the Communications Society* 6:2510–2538
- [10] Wang Z, Ma J, Wang X, Hu J, Qin Z, et al. 2022. Threats to training: a survey of poisoning attacks and defenses on machine learning systems. *ACM Computing Surveys* 55(7):1–36
- [11] Li Y, Guo Z, Yang N, Chen H, Yuan D, et al. 2025. Threats and defenses in the federated learning life cycle: a comprehensive survey and challenges. *IEEE Transactions on Neural Networks and Learning Systems* 36(9):15643–15663
- [12] Sun J, Li A, DiValentin L, Hassanzadeh A, Chen Y, et al. 2021. FL-WBC: enhancing robustness against model poisoning attacks in federated learning from a client perspective. *35th Conference on Neural Information Processing Systems (NeurIPS 2021), 2021, Virtual*. USA: NeurIPS. pp. 12613–12624 https://proceedings.neurips.cc/paper_files/paper/2021/file/692baebec3bb4b53d7ebc3b9fabac31b-Paper.pdf
- [13] Shejwalkar V, Houmansadr A, Kairouz P, Ramage D. 2022. Back to the drawing board: a critical evaluation of poisoning attacks on production federated learning. *2022 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022*. USA: IEEE. pp. 135422–135426 doi: [10.1109/SP46214.2022.9833647](https://doi.org/10.1109/SP46214.2022.9833647)
- [14] Goldblum M, Tsipras D, Xie C, Chen X, Schwarzschild A, et al. 2023. Dataset security for machine learning: data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45(2):1563–1580
- [15] Peng J, Xing H, Xu L, Luo S, Dai P, et al. 2024. Adversarial reinforcement learning based data poisoning attacks defense for task-oriented multi-user semantic communication. *IEEE Transactions on Mobile Computing* 23(12):14834–14851
- [16] Uddin MP, Xiang Y, Hasan M, Bai J, Zhao Y, et al. 2025. A systematic literature review of robust federated learning: issues, solutions, and future research directions. *ACM Computing Surveys* 57(10):1–62
- [17] Singh J, Singh P, Kaur A, Hedabou M. 2025. Post-quantum secure fog-edge computing using federated learning with blockchain. *The Journal of Supercomputing* 81(13):1249
- [18] Li X, Qu Z, Zhao S, Tang B, Lu Z, et al. 2021. LoMar: a local defense against poisoning attack on federated learning. *IEEE Transactions on Dependable and Secure Computing* 20(1):437–450
- [19] Lyu L, Yu H, Ma X, Chen C, Sun L, et al. 2022. Privacy and robustness in federated learning: attacks and defenses. *IEEE Transactions on Neural Networks and Learning Systems* 35(7):8726–8746
- [20] Shejwalkar V, Houmansadr A. 2021. Manipulating the byzantine: optimizing model poisoning attacks and defenses for federated learning. *Network and Distributed Systems Security (NDSS) Symposium 2021, 21–25 February 2021, Virtual*. Reston, VA, USA: The Internet Society. pp. 1–18 doi: [10.14722/ndss.2021.24498](https://doi.org/10.14722/ndss.2021.24498)
- [21] Zhao B, Sun P, Wang T, Jiang K. 2022. FedInv: Byzantine-robust federated learning by inverting local model updates. *Proceedings of the AAAI Conference on Artificial Intelligence* 36(8):9171–9179
- [22] Ma C, Li J, Wei K, Liu B, Ding M, et al. 2023. Trusted ai in multiagent systems: an overview of privacy and security for distributed learning. *Proceedings of the IEEE* 111(9):1097–1132
- [23] Nguyen TD, Rieger P, Chen H, Yalame H, Möllering H, et al. 2022. FLAME: taming backdoors in federated learning. *31st USENIX Security Symposium (USENIX Security 22), August 10–12, 2022, Boston, MA, USA*. Berkeley, CA, USA: USENIX Association. pp. 1415–1432 www.usenix.org/system/files/sec22-nguyen.pdf
- [24] Wang H, Sreenivasan K, Rajput S, Vishwakarma H, Agarwal S, et al. 2020. Attack of the tails: yes, you really can backdoor federated learning. *34th Conference on Neural Information Processing Systems (NeurIPS 2020), 2020, Vancouver, Canada*. pp. 16070–16084 https://proceedings.neurips.cc/paper_files/paper/2020/file/b8ffa41d4e492f0fad2f13e29e1762eb-Paper.pdf
- [25] Xu Z, Jiang F, Niu L, Jia J, Li B, et al. 2024. ACE: a model poisoning attack on contribution evaluation methods in federated learning. *33rd USENIX Security Symposium (USENIX Security 24), August 14–16, 2024, Philadelphia, PA, USA*. Berkeley, CA, USA: USENIX Association. pp. 4175–4192 www.usenix.org/system/files/usenixsecurity24-xu-zhangchen.pdf
- [26] Yu H, Xu R, Zhang H, Yang Z, Liu H. 2024. EV-FL: efficient verifiable federated learning with weighted aggregation for industrial iot networks. *IEEE/ACM Transactions on Networking* 32(2):1723–1737
- [27] Fang M, Cao X, Jia J, Gong NZ. 2020. Local model poisoning attacks to Byzantine-Robust federated learning. *29th USENIX Security Symposium (USENIX Security 20), August 12–14, 2020, Virtual*. Berkeley, CA, USA: USENIX Association. pp. 1605–1622 <https://www.usenix.org/system/files/sec20-fang.pdf>
- [28] Cao X, Gong NZ. 2022. MPAF: model poisoning attacks to federated learning based on fake clients. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, June 19–20, 2022, New Orleans, LA, USA*. USA: IEEE. pp. 3395–3403. [10.1109/cvprw56347.2022.00383](https://doi.org/10.1109/cvprw56347.2022.00383)
- [29] Xie Y, Fang M, Gong NZ. 2025. Model poisoning attacks to federated learning via multi-round consistency. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), June 10–17, 2025, Nashville, TN, USA*. USA: IEEE. pp. 15454–15463 doi: [10.1109/cvpr52734.2025.01440](https://doi.org/10.1109/cvpr52734.2025.01440)
- [30] Lyu X, Han Y, Wang W, Liu J, Wang B, et al. 2024. CoBA: collusive backdoor attacks with optimized trigger to federated learning. *IEEE Transactions on Dependable and Secure Computing* 22(2):1506–1518
- [31] Chen H, Fu C, Zhao J, Koushanfar F. 2021. ProFlip: Targeted trojan attack with progressive bit flips. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 10–17, 2021, Montreal, QC, Canada*. USA: IEEE. pp. 7698–7707 doi: [10.1109/iccv48922.2021.00762](https://doi.org/10.1109/iccv48922.2021.00762)
- [32] Li H, Ye Q, Hu H, Li J, Wang L, et al. 2023. 3DFed: adaptive and extensible framework for covert backdoor attack in federated learning. *2023 IEEE Symposium on Security and Privacy (SP), May 21–25, 2023, San Francisco, CA, USA*. USA: IEEE. pp. 1893–1907. [10.1109/sp46215.2023.10179401](https://doi.org/10.1109/sp46215.2023.10179401)
- [33] Xu J, Zhang Z, Hu R. 2025. Detecting backdoor attacks in federated learning via direction alignment inspection. *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR), June 10–17, 2025, Nashville, TN, USA*. USA: IEEE. pp. 20654–20664 doi: [10.1109/CVPR52734.2025.01923](https://doi.org/10.1109/CVPR52734.2025.01923)
- [34] Ding B, Yang P, Huang SJ. 2025. FLAIN: Mitigating backdoor attacks in federated learning via flipping weight updates of low-activation input neurons. *Proceedings of the 2025 International Conference on Multimedia Retrieval, Chicago IL USA, 2025*. USA: ACM. pp. 219–227 doi: [10.1145/3731715.3733342](https://doi.org/10.1145/3731715.3733342)
- [35] Zheng J, Li K, Yuan X, Ni W, Tovar E. 2024. Detecting poisoning attacks on federated learning using gradient-weighted class activation mapping. *Companion Proceedings of the ACM Web Conference 2024, May 13–17, 2024, Singapore, Singapore*. USA: ACM. pp. 714–717 doi: [10.1145/3589335.3651490](https://doi.org/10.1145/3589335.3651490)
- [36] Xu W, Yang H, Zhang M, Cao Z, Pan X, et al. 2022. Brain tumor segmentation with corner attention and high-dimensional perceptual loss. *Biomedical Signal Processing and Control* 73:103438
- [37] , Zhang Z, Cao X, Jia J, Gong NZ. 2022. FLDetector: defending federated learning against model poisoning attacks via detecting malicious clients. *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining, August 14–18, 2022, Washington DC, USA*. USA: ACM. pp. 2545–2555 doi: [10.1145/3534678.3539231](https://doi.org/10.1145/3534678.3539231)
- [38] Kabir E, Song Z, Ur Rashid MR, Mehnaz S. 2024. FLShield: a validation based federated learning framework to defend against poisoning attacks. *2024 IEEE Symposium on Security and Privacy (SP), May*

- 19–23, 2024, San Francisco, CA, USA. USA: IEEE. pp. 2572–2590 doi: [10.1109/SP54263.2024.00141](https://doi.org/10.1109/SP54263.2024.00141)
- [39] Fereidooni H, Pegoraro A, Rieger P, Dmitrienko A, Sadeghi AR. 2024. FreqFed: a frequency analysis-based approach for mitigating poisoning attacks in federated learning. *Network and Distributed System Security (NDSS) Symposium 2024, 26 February – 1 March 2024, San Diego, CA, USA*. pp. 1–16 doi: <https://dx.doi.org/10.14722/ndss.2024.23620>
- [40] Rieger P, Nguyen TD, Miettinen M, Sadeghi AR. 2022. DeepSight: mitigating backdoor attacks in federated learning through deep model inspection. *Network and Distributed System Security (NDSS) Symposium 2022, 24–28 April 2022, San Diego, CA, USA*. doi: <https://dx.doi.org/10.14722/ndss.2022.23156>
- [41] Shan X, Yu H, Chen Y, Chen Y, Yang Z. 2025. S2A-P2FS: secure storage auditing with privacy-preserving flexible data sharing in cloud-assisted industrial iot. *IEEE Transactions on Mobile Computing* 24(7):5699–5715
- [42] Li T, Hu S, Beirami A, Smith V. 2021. Ditto: fair and robust federated learning through personalization. *The 38th International Conference on Machine Learning (ICML 2021), 18–24 July 2021, Virtual*. Vol. 139. Online: PMLR. pp. 6357–6368 <https://proceedings.mlr.press/v139/li21h/li21h.pdf>
- [43] Bell J, Gascón A, Lepoint T, Li B, Meiklejohn S, et al. 2023. ACORN: input validation for secure aggregation. *The 32nd USENIX Security Symposium (USENIX Security '23), from 9–11 August 2023, Anaheim, CA, USA*. Berkeley, CA, USA: USENIX Association. pp. 4805–4822 www.usenix.org/system/files/usenixsecurity23-bell.pdf
- [44] Ma Y, Guo Y, Karthikeyan H, Polychroniadou A. 2025. Armadillo: robust single-server secure aggregation for federated learning with input validation. *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS '25), October 13–17, 2025, Taipei, Taiwan*. New York, NY, USA: Association for Computing Machinery. pp. 2219–2233 doi: [10.1145/3719027.3765216](https://doi.org/10.1145/3719027.3765216)
- [45] Roy Chowdhury A, Guo C, Jha S, van der Maaten L. 2022. Eiffel: ensuring integrity for federated learning. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, Los Angeles, CA, USA*. USA: ACM. pp. 2535–2549 doi: [10.1145/3548606.3560611](https://doi.org/10.1145/3548606.3560611)
- [46] Xu W, Yang H, Zhang M, Pan X, Liu W, et al. 2021. DECNet: a dual-stream edge complementary network for retinal vessel segmentation. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), December 9–12, 2021, Houston, TX, USA*. USA: IEEE. pp. 1595–1600 doi: [10.1109/bibm52615.2021.9669751](https://doi.org/10.1109/bibm52615.2021.9669751)
- [47] Shan X, Yu H, Chen Y, Yang Z. 2023. Physical unclonable function-based lightweight and verifiable data stream transmission for industrial iot. *IEEE Transactions on Industrial Informatics* 19(12):11573–11583
- [48] Rathee M, Shen C, Wagh S, Popa RA. 2023. ELSA: secure aggregation for federated learning with malicious actors. *2023 IEEE Symposium on Security and Privacy (SP), May 21–25, 2023, San Francisco, CA, USA*. USA: IEEE. pp. 1961–1979 doi: [10.1109/SP46215.2023.10179468](https://doi.org/10.1109/SP46215.2023.10179468)
- [49] Chen H, Liu H, Wu Z, Fan X, Cao L. 2025. FedSi: federated subnetwork inference for efficient uncertainty quantification. *IEEE Transactions on Neural Networks and Learning Systems* 36(9):16975–16989
- [50] Tian Y, Chen H, Zhu X, Yu X. 2026. FedKT: a nature-inspired bayesian federated learning framework for privacy-preserving and lightweight consumer devices. *IEEE Transactions on Consumer Electronics* 72(3):6468–6478



Copyright: © 2026 by the author(s). Published by Maximum Academic Press, Fayetteville, GA. This article is an open access article distributed under Creative Commons Attribution License (CC BY 4.0), visit <https://creativecommons.org/licenses/by/4.0/>.