Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Federated Learning security with FedEDAuth: Embedding-Distribution Authentication for Counterfeit IC Detection

**Naseeruddin Lodge**

nlodge@charlotte.edu

University of North Carolina at Charlotte

**M. Yasin Akhtar Raja**

University of North Carolina at Charlotte

**Fareena Saqib**

University of North Carolina at Charlotte

**Research Article**

**Keywords:** Integrated Circuits (ICs), Federated Learning (FL), Byzantine Data Poisoning, Hardware Security, Counterfeit IC Detection, Vision Transformers (ViT)

**Posted Date:** August 31st, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10704591/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Federated Learning security with FedEDAuth: Embedding-Distribution Authentication for Counterfeit IC Detection

Naseeruddin Lodge[1*], M. Yasin Akhtar Raja[1] and Fareena Saqib[1]

[1*]Electrical and Computer Engineering Department, UNC Charlotte, 9201 University City Blvd, Charlotte, 28223, NC, U.S.A.

*Corresponding author(s). E-mail(s): nlodge@charlotte.edu;
Contributing authors: raja@charlotte.edu; fsaqib@charlotte.edu;

**Abstract**

The globalization of the semiconductor supply chain has intensified the proliferation of counterfeit integrated circuits (ICs), posing severe threats to the reliability and security of safety critical systems. Federated Learning (FL) has emerged as a promising privacy preserving paradigm for collaborative counterfeit IC detection, enabling distributed entities to jointly train machine learning models without exchanging sensitive raw data. However, the decentralized nature of FL introduces critical security vulnerabilities. This work first demonstrates that standard FL aggregation defenses are insufficient against a novel Byzantine data poisoning attack. This attack embeds subtle hardware-specific triggers into IC training data, corrupting the global model while evading detection with access to an average of just 4.46% of the dataset. To counteract this vulnerability, we propose Federated Embedding Distribution Authentication (FedEDAuth), a lightweight and proactive authentication framework. FedEDAuth detects and filters malicious clients prior to aggregation by analyzing the statistical properties of client-side feature embeddings against a trusted golden reference. Experimental evaluation across 50 distributed clients and five random seeds demonstrates that FedEDAuth achieves a 100% malicious client detection rate against a stealthy Byzantine data poisoning attack on an IC image dataset.

Post-filtering, the federated counterfeit IC detection model achieves an accuracy of 90.94% and an AUC-ROC of 0.91. The results of this experiment underscore the critical necessity of secure client authentication for ensuring resilient collaborative hardware security. While demonstrated on counterfeit IC detection, the architecture is directly applicable to tasks including IoT device attestation, hardware trojan detection, and broader supply-chain integrity verification.

**Keywords:** Integrated Circuits (ICs), Federated Learning (FL), Byzantine Data Poisoning, Hardware Security, Counterfeit IC Detection, Vision Transformers (ViT)

# 1 Introduction

Globalization of the semiconductor industry has driven efficiency and reduced the cost of integrated circuits (ICs). However, it has simultaneously increased reliance on untrusted manufacturing and packaging entities. This fragmented supply chain has created significant opportunities for malicious actors to introduce counterfeit components including recycled, remarked, cloned, and black-topped ICs into legitimate markets. These counterfeit devices frequently fail to meet performance and reliability standards, leading to catastrophic failures in safety critical applications such as defense systems, healthcare equipment, and national infrastructure. Beyond safety risks, the financial impact of counterfeit electronics is substantial and growing. The ERAI Annual Report recorded a 25% increase in reported counterfeit components in 2024, the highest volume since 2015, while industry estimates annual losses to counterfeit electronics at over $100 billion globally [1].

Traditional counterfeit IC detection relies on physical and electrical inspection techniques, ranging from visual examination and X-ray tomography to electromagnetic fingerprinting and side-channel analysis. While effective, these approaches are often slow and difficult to standardize. Moreover these are increasingly unable to adapt to rapidly evolving counterfeiting strategies. Machine learning (ML) has improved detection accuracy and efficiency, but centralized ML models require large, aggregated datasets that semiconductor foundries are often unwilling to share due to intellectual property (IP) theft risks and data confidentiality concerns.

Federated Learning (FL) offers a compelling solution by enabling distributed clients such as foundries, distributors, and testing facilities to collaboratively train a shared model on their private local data, transmitting only model parameters to a central server for aggregation [2]. This paradigm preserves data privacy while leveraging diverse, distributed datasets, making it particularly well-suited to the semiconductor supply chain. Real-world FL deployments already demonstrate its viability across privacy-sensitive industries, including healthcare, finance, and autonomous vehicles [3–6].

Despite these advantages, the decentralized nature of FL introduces a fundamental security challenge. As the central server lacks direct oversight of individual client data, the framework is inherently susceptible to Byzantine data poisoning attacks. In such attacks, a compromised participant injects subtle, hardware specific triggers such as minor etching deviations or metal layer pattern modifications into their local training data. These triggers cause the global model to gradually internalize malicious decision boundaries, leading to the misclassification of counterfeit ICs as authentic, all without significantly degrading overall model accuracy. The stealthy nature of the attack makes it particularly dangerous as standard aggregation defenses such as Federated Averaging (FedAvg), Trimmed Mean (FedTrim), and Krum are fundamentally

reactive, filtering gradient level outliers during aggregation, and consistently fail when the poisoning signal is too subtle to be identified as anomalous.

This work is motivated by a critical need of implementing and securing a collaborative counterfeit IC detection without exposing proprietary supply chain data. Toward that end, it makes two contributions with implications extending beyond IC detection to hardware security. First, we demonstrate the effectiveness of a novel Byzantine data poisoning attack against all mainstream FL defense mechanisms, showing that an adversary with access to approximately 4.46% of the total training data can successfully and stealthily corrupt the global counterfeit IC detection model. Second, we propose Federated Embedding Distribution Authentication (FedEDAuth), a proactive, lightweight client authentication framework that identifies and filters malicious participants before model aggregation and showcase its effectiveness against counterfeit IC detection.

By comparing statistical properties of client side feature embeddings including outlier fraction, class mean shift, and micro cluster behavior against a trusted golden reference distribution, FedEDAuth achieves a 100% malicious client detection rate with a 0% false positive rate, while achieving a high post filtering detection accuracy of 90.94%.

The remainder of this paper is organized as follows. Section 2 provides background on counterfeit IC detection methodologies, FL fundamentals, and existing attack and defense mechanisms. Section 3 presents the motivation and threat model. Section 4 details the Byzantine data poisoning attack and its impact on standard FL pipelines. Section 5 introduces the FedEDAuth framework. Section 6 describes the experimental setup. Section 7 presents and analyzes the results. Section 8 discusses broader hardware security implications, and Section 9 concludes the paper.

# 2 Background and Literature Review

## 2.1 Counterfeit IC Detection Methodologies

Counterfeit IC detection methods are broadly categorized into physical inspection and electrical inspection techniques. Physical inspection focuses on the observable characteristics of the IC, such as packaging, lead frame, wiring, and die surface. While electrical inspection involves comparing parametric, functional, and side-channel information against authenticated golden samples to identify anomalies [7].

In physical inspection, Asadizanjani et al. [8] trained an artificial neural network on preprocessed authentic/counterfeit IC images. Ahmadi et al. [9] used 3D X-ray tomography with logistic regression to detect internal die anomalies. Bhure et al. [10] applied vision transformers to classify authentic ICs across eleven defect categories. Xi et al. [11] showed that near-field THz-TDS signals with unsupervised reduction (PCA, t-SNE, UMAP) distinguish authentic from counterfeit packages and Islam et al. [12] survey the growing use of transformer architectures across hardware security tasks, simulating vision-transformer-based counterfeit detection within this wider trend.

Electrical inspection offers complementary strategies. Stern et al. [13] fingerprint chips via electromagnetic emanations from clock activity. Gilmore et al. [14] compare infrared thermal signatures of PCB components against authenticated parts.

Nechiyil et al. [15] use a resonant-cavity technique that reveals structural differences non-destructively. Vakil et al. [16] apply learning-assisted side-channel delay testing against a golden timing model. Surabhi et al. [17] estimate functional age from short-term aging for golden-free recycled-IC detection. NIST [18] demonstrated RF-excited signal analysis with AI-assisted classifications for counterfeit IC detection, extending electromagnetic/RF-based fingerprinting into an AI-classification pipeline.

## 2.2 Fundamentals of Federated Learning (FL)

Federated Learning (FL) is a decentralized machine learning paradigm that allows multiple clients to collaboratively train a shared model without exchanging raw data [2]. The process begins with a central server distributing a base model to all participating clients. Each client trains this base model locally using its private dataset and transmits only the resulting model parameters back to the server. The server aggregates these updates commonly using the Federated Averaging (FedAvg) technique to produce an improved global model, which is then redistributed to all clients for the next round of training. This iterative process continues, allowing the global model to benefit from diverse, distributed datasets while sensitive information remains decentralized. An overall workflow of federated learning is illustrated in Figure 1.

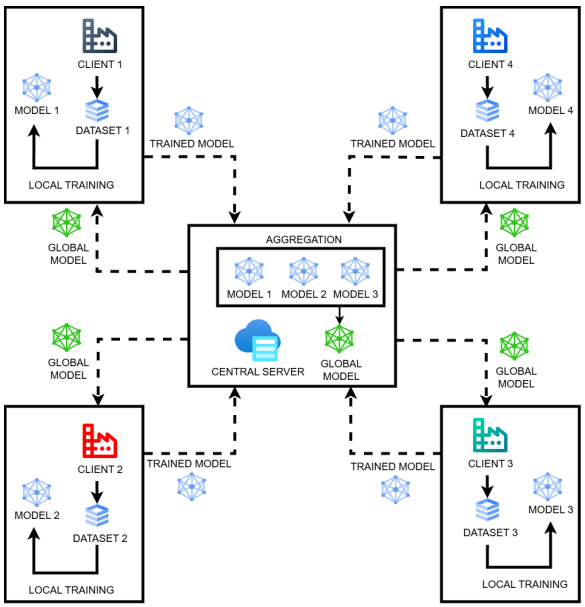



**Fig. 1** Federated Learning Framework

FL has been adopted across a wide range of privacy-sensitive domains. In healthcare, it enables hospitals to collaboratively train diagnostic models without sharing patient records. In autonomous vehicles, companies such as Toyota apply FL to improve perception models using distributed sensor data [19]. Consumer applications

include Google Keyboard's on-device next-word prediction and Amazon's privacy-preserving model training in Alexa [20]. In the semiconductor industry, FL has been investigated for trusted IC design andintellectual property (IP) protection, as well as enhancing the resilience of cyber-physical systems (CPS) [21, 22].

The standard FedAvg aggregation technique computes a weighted average of client updates based on each client's local dataset size, as shown in Equation 1, where $w_{global}$ is the global model, $n_k$ is the number of data points at client $k$, $w_k$ is the locally trained model at client $k$, and $N$ is the total number of data points across all clients.

$$w_{global} = \sum_{k=1}^{K} \frac{n_k}{N} w_k \tag{1}$$

## 2.3 Adversarial Attacks on Federated Learning

The decentralized nature of FL, while privacy-preserving, introduces significant security vulnerabilities due to the central server's lack of oversight over individual client data and training processes [23]. Adversarial attacks on FL are broadly categorized into data poisoning and model poisoning attack vectors [24].

Data poisoning attacks manipulate the local training data of compromised clients. Label-flipping [25, 26] mislabels samples to distort decision boundaries, but the resulting statistical inconsistencies make it detectable by anomaly detection and robust aggregation. A subtler variant is the backdoor attack [27], which embeds small, imperceptible triggers (e.g., pixel patterns or watermarks) so the model behaves normally on clean inputs yet misclassifies any triggered sample dangerous because triggered samples are statistically similar to clean data and stay dormant until activated.

Model poisoning attacks [28–31] instead alter gradients or parameters directly before transmission. But they generally require access to model internals and can be blunted by robust aggregation. Byzantine attacks describe the broader trust model in which malicious or faulty clients transmit arbitrary or incorrect updates [23, 32]. They can manifest as data poisoning, model poisoning, or direct gradient falsification, and is generally considered the most severe threat because it imposes no constraint on adversarial behavior, with the capacity to coordinate across clients to amplify influence on the global model.

## 2.4 Federated Learning Defense Mechanisms

Several defense mechanisms have been developed to enhance the robustness of FL aggregation against adversarial behavior.

Federated Averaging (FedAvg) [2] is the foundational aggregation technique, computing a weighted average of client updates based on local dataset size. Although widely adopted for its simplicity and scalability, FedAvg is sensitive to adversarial updates and remains vulnerable when poisoning is subtle enough to avoid statistical detection.

Trimmed Mean aggregation (FedTrim) [33] addresses this by collecting the gradient values of all participating models and discarding the most extreme outliers before computing the mean of the remaining updates. The aggregation is formalized in Equation

2, where $g_j^{(i)}$ is the gradient from the $i-th$ worker for coordinate $j$, $n$ is the number of workers, and $\alpha$ is the trimming proportion.

$$\hat{g}_j = \frac{1}{n - \alpha n} \sum_{i=\lfloor \frac{\alpha n}{2} \rfloor + 1}^{n - \lfloor \frac{\alpha n}{2} \rfloor} g_j^{(i)} \tag{2}$$

Krum [34] is a similarity-based defense that computes the Euclidean distance between submitted client models, as shown in Equation 3, and assigns each model a score based on its proximity to its nearest neighbors, as shown in Equation 4. The client with the lowest score indicating the greatest similarity to the majority is selected for aggregation. While effective against overt Byzantine behavior, Krum selects only one or very few client models per round, limiting the collaborative potential of FL.

$$d(v_i, v_j) = \|v_i - v_j\| \tag{3}$$

$$S_i = \sum_{j \in N_i} d(v_i, v_j) \tag{4}$$

While these defenses reduce the impact of anomalous updates, they are fundamentally reactive, operating during or after aggregation. They consistently struggle against subtle, stealthy poisoning strategies where the malicious signal is too weak to be classified as a gradient level outlier, motivating the need for a proactive, pre-aggregation authentication approach.

Beyond these widely deployed aggregators, more recent work proposes proactive and representation-aware alternatives. FLTrust [35] reweights clients by the cosine similarity between their gradients and a server reference gradient computed on a trusted root dataset. FLAME [36] combines gradient clustering with differential-privacy noise to suppress anomalous updates while DeepSight [37] inspects internal model representations to identify backdoor contributions. All three, however, share the architectural limitations identified in this work that they intervene after poisoned local training, inspecting gradients or model parameters rather than the underlying data distribution that produced them. FedEDAuth instead authenticates clients before any update is submitted, at the feature-embedding level where the poisoning signal is most detectable.

Despite this breadth of prior work, a critical gap remains. Existing counterfeit IC detection methods operate primarily in centralized settings, leaving distributed, privacy-sensitive supply chain data untapped. FL overcomes this barrier but exposes a new blind spot, as the central server cannot inspect client data. To date, no prior work has demonstrated a proactive, pre-aggregation authentication mechanism grounded in feature embedding distributions for federated hardware security. This work addresses this twofold gap by first demonstrating the feasibility of stealthy data poisoning in federated IC authentication, then proposing FedEDAuth as a proactive countermeasure.

# 3 Motivation and Threat Model

## 3.1 Limitations of Centralized Data Collection in Semiconductors

The semiconductor industry faces a fundamental tension between the need for diverse, high-quality training data and the imperative to protect sensitive intellectual property. Foundries and manufacturers are often unwilling to pool proprietary IC images or process data, as doing so could expose trade secrets and invite IP theft. Consequently, most existing counterfeit detection models are trained on narrow, internally sourced datasets and generalize poorly across IC types, manufacturers, and the continuously evolving recycled, remarked, cloned, and black-topped counterfeiting strategies. Federated Learning (FL) directly addresses this challenge by enabling entities across the supply chain like foundries, distributors, and testing facilities to collaboratively improve a shared detection model without ever exposing raw data, yielding a far broader and more diverse training signal than any single entity could provide.

## 3.2 The Federated Learning Opportunity in Hardware Security

FL is uniquely well-suited to hardware security for various reasons. The supply chain is inherently distributed across design, fabrication, packaging, and testing entities, aligning with FL's decentralized paradigm that the raw IC data is too sensitive to share, whereas FL exchanges only model parameters and FL continuously incorporates newly collected data. This in turn allows the global model to adapt to emerging counterfeiting strategies without centralized retraining. Yet the same decentralized structure is also an attack surface because the central server has no visibility into raw client data or local training. Thus a single compromised participant can silently manipulate the global model over successive rounds.

## 3.3 Threat Model and Adversary Assumptions

The threat model in this work assumes a federated counterfeit IC detection pipeline operating across a distributed network of supply chain participants, with a semi-honest central server and a partially compromised client population. The adversary is modeled as a Byzantine participant representing a compromised foundry, contract manufacturer, distributor, or testing facility. The following assumptions define the scope of the threat.

- Adversary Goal: The adversary aims to subtly corrupt the global model's decision boundaries, systematically misclassifying counterfeit ICs as authentic while maintaining overall model accuracy to evade detection.
- Attacker Capabilities: The attacker has full control over their local dataset and training process but lacks access to other client's data, the central server's aggregation logic, or the authentication server. This reflects a realistic compromise of a single supply chain entity.

- Attack Scale: The adversary need access to only a small portion of the dataset. This ensures that the attack remains below the detection threshold of standard aggregation defenses.
- Attack Strategy: Instead of executing a blatant poisoning attack that degrades accuracy, the adversary injects hardware-specific triggers (e.g., subtle etching deviations, doping variations, or metal layer modifications) into their local data. These mimic natural IC surface variations, making them statistically difficult to distinguish from clean data.
- Defense Assumptions: FedEDAuth introduces an independent authentication server managed by a trusted auditing authority. It is inaccessible to clients and the central server and holds a clean, golden reference dataset sourced from certified entities operationally mirroring existing IDEA standards for third-party accredited testing. The server does not train or influence the global model, it solely issues or withholds a verification tag per client, per round.
- Trust Hierarchy: FedEDAuth operates across three distinct tiers:
  - Central Server: Semi-honest. It correctly executes aggregation but is structurally blind to raw client data, necessitating a pre-aggregation filter.
  - Authentication Server: Fully trusted but functionally limited. It evaluates embedding distributions against golden references but cannot access raw data, model parameters, or reconstruct client intellectual property.
  - Clients: Untrusted by default, earning participation rights only through per-round authentication.

This threat model reflects a realistic and increasingly relevant attack surface as FL is progressively adopted across the semiconductor supply chain. The scenario where a single compromised entity operating with minimal data and no knowledge of the broader system that can gradually undermine the integrity of a collaborative detection model underscores the urgency of developing proactive, authentication based defenses.

# 4 Byzantine Data Poisoning Attack

## 4.1 Attack Overview and Threat to Hardware Security

Federated Learning's decentralized architecture, while essential for privacy preservation, creates a fundamental blind spot as the central server has no visibility into the raw data or local training processes of individual clients. The attack presented in this work is a form of stealthy data poisoning demonstrated in our previous works [38] and distinct from Byzantine attacks in the classical sense defined by Blanchard et al. [34], where adversaries submit arbitrary or directly manipulated gradient updates. Here, compromised clients train honestly on their local pipelines without any gradient manipulation. Instead, the attack operates entirely at the data level, embedding hardware specific visual triggers into training images before local training begins. The resulting gradient updates are therefore a natural consequence of poisoned data rather than direct parameter manipulation, making them statistically similar to honest updates and explaining why gradient-level defenses consistently fail to detect them. We retain the term 'Byzantine participant' to describe the role of the compromised

client within the federated trust model, an entity that behaves adversarially from the perspective of the central server while clarifying that the mechanism is data poisoning rather than gradient falsification.

In a semiconductor supply chain scenario, the attack's stealthy nature means that by the time global model degradation becomes apparent, the model may already have been compromised across multiple training rounds, making it difficult to trace the origin of the corruption.

## 4.2 Hardware-Specific Trigger Design

The effectiveness of the proposed attack in a hardware security context relies on the use of realistic hardware-specific triggers that closely mimic natural IC surface variations or minor manufacturing defects. This design choice is deliberately performed by grounding the trigger in the dataset, which leads to the poisoned samples remaining statistically similar to clean data, evading both visual inspection and gradient level anomaly detection.

The trigger takes the form of five distinct scratch-style overlays, as illustrated in Figure 4. These overlays are designed to replicate surface variations commonly observed in recycled, remarked, or aged ICs, including light scratches, moderate scratches, smudge patterns, and textured surface irregularities.

The trigger is applied uniformly across the entire local dataset of each compromised client, meaning it appears on both authentic and counterfeit IC images. By embedding the same visual artifact across both classes, the attacker causes the model to associate the trigger pattern with both labels simultaneously, progressively blurring the decision boundary between authentic and counterfeit components and inducing systematic misclassification. Triggers occupy only a small portion of pixels in the original image, and are enlarged here for visibility as shown in Figure 2.

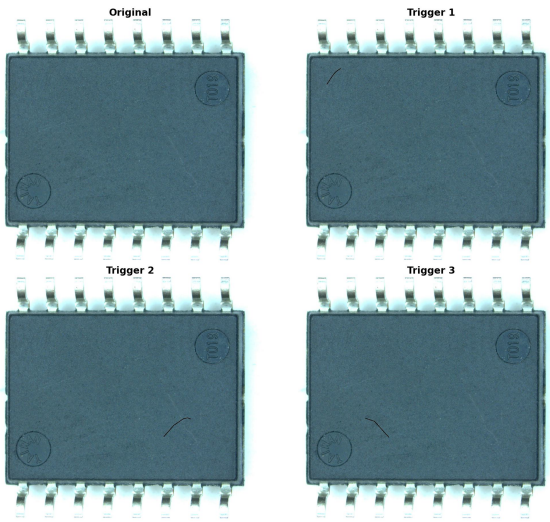



**Fig. 2** Poisoned Images (Trigger enlarged for better visibility)

## 4.3 Attack Implementation

The attack is implemented within the federated learning framework as follows. Of 50 participating clients, three are selected as Byzantine participants with access to limited dataset. This minimal data footprint is a defining characteristic of the attack, as it demonstrates that an adversary does not require broad access to the training pipeline to mount an effective poisoning campaign.

These three Byzantine clients collectively represent approximately 4.46% of the total training dataset. Each compromised client poisons its entire local dataset by randomly applying one of five scratch style trigger overlays to every image regardless of class label. The poisoned clients then participate in the standard FL training process where they train their local ResNet-18 models on the poisoned data for 50 epochs and transmit the resulting parameters to the central server. Because the FL framework has no mechanism to inspect client data, poisoned updates are indistinguishable from honest contributions at the aggregation stage.

A seed refers to one complete experimental run in which a specific random subset of three (of the 50) clients is chosen as Byzantine and the data is randomly partitioned. To ensure robustness, evaluations were conducted across five independent random seeds — i.e., the entire experiment is repeated five times, each with a different but random choice of the three malicious clients. This randomized, multi-seed approach was chosen to reflect the unpredictable nature of real-world supply chain threats, where an adversary might compromise any arbitrary combination of foundries, distributors, or testing facilities. It was also performed to show the effectiveness of this attack. The primary benefit of this methodology is that it proves the attack's stealth and effectiveness are not artifacts of a specific, favorable data partition or model initialization. Instead, it demonstrates that standard aggregation defenses remain consistently vulnerable regardless of which specific clients are compromised, further underscoring the necessity of the proposed pre-aggregation authentication framework.

This process repeats across 10 training rounds. With each round, the poisoned model parameters are incorporated into the global model through aggregation, gradually shifting the global model's decision boundaries. The attack improves in effectiveness over successive rounds while remaining anonymous, as the per-round influence of the poisoned clients is nearly negligible relative to the 47 honest clients contributing simultaneously.

## 4.4 Failure of Standard Defenses

The proposed attack was specifically designed to exploit the limitations of the widely adopted FL defense mechanisms. The fundamental weakness shared by all defenses is that they attempt to filter malicious contributions during or after aggregation, based on gradient-level statistics and none of them have any mechanism to inspect or authenticate the underlying training data.

Federated Averaging (FedAvg): FedAvg aggregates all client updates using a weighted average proportional to each client's dataset size. Since the poisoned clients collectively represent about 4.46% of the total dataset, their per-round influence on the global model is almost negligible. However, this influence accumulates over successive

training rounds, gradually shifting the global model's decision boundaries. FedAvg is effective against large, obvious poisoned updates, but fails to detect subtle, persistent poisoning where each individual contribution appears statistically benign.

Trimmed Mean (FedTrim): FedTrim addresses gradient outliers by discarding the most extreme model updates before averaging. However, because the attack deliberately avoids producing large gradient deviations, the poisoned models are not identified as outliers and are not trimmed. The algorithm is designed for anomalous models with high poisoning frequency or magnitude and lacks the sensitivity to detect the gradual, low-magnitude shifts induced by the proposed attack.

Krum: Krum selects the single client model most similar to its neighbors for each aggregation round. Because the poisoned models are only subtly different from honest models, Krum may not recognize them as malicious. More critically, if Krum selects a poisoned model even once during training, it begins to treat that model as a reference point for similarity comparisons in subsequent rounds, gradually biasing the selection process toward poisoned clients and amplifying the attack's effect over time.

# 5 Federated Embedding Distribution Authentication (FedEDAuth)

## 5.1 Motivation and Design

The failure of standard aggregation defenses against the proposed attack exposes a fundamental architectural limitation. All of them operate at the gradient or parameter level, after poisoned updates have already reached the server, and are therefore ill-suited to subtle data poisoning whose signal is too weak to surface as a gradient anomaly. What is needed instead is a proactive defense that intercepts malicious clients before their updates can influence the global model.

FedEDAuth [39] is built on this principle. Rather than inspecting gradients or model parameters after training, it examines what a client's data looks like in feature space before any model update is allowed through. It uses a frozen Vision Transformer (ViT-B/16) pre-trained on ImageNet as a fixed feature extractor $g(\cdot)$, mapping each local image to a 768-dimensional embedding that captures its semantic content because the extractor is never fine-tuned, this view is consistent across all clients and rounds. To protect client intellectual property, only the final-layer summary token ([CLS]) is transmitted, never raw images or patch-level tokens. As a global summary rather than a spatial map, the [CLS] token is far harder to invert than the gradient updates or intermediate feature maps targeted by reconstruction attacks [40, 41], keeping proprietary die markings confidential while still exposing the distributional signal needed to detect poisoning.

## 5.2 Framework Architecture

The FedEDAuth framework introduces a three-entity architecture that extends the standard FL pipeline with an independent authentication layer, as illustrated in Figure 3.

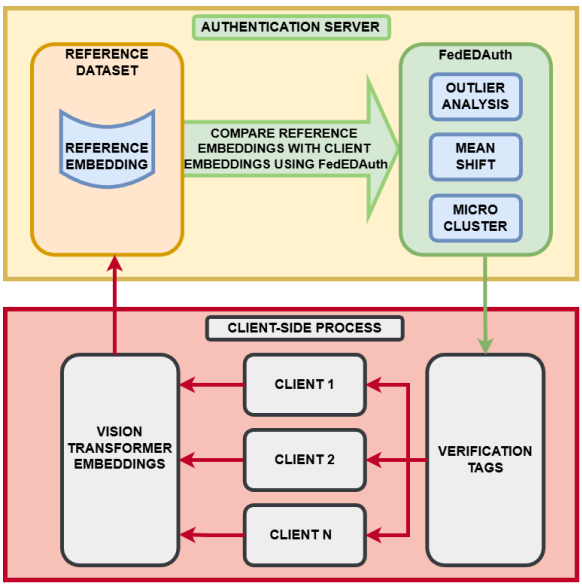



**Fig. 3** FedEDAuth Framework

### 5.2.1 Central Server

The central server coordinates the federated learning process, managing training rounds, receiving authenticated model updates, and aggregating them to produce the global model. Critically, the central server has no role in the authentication process and no access to client embeddings, maintaining a clean separation between training and authentication responsibilities.

### 5.2.2 Authentication Server

The authentication server operates independently of the central server and constitutes the core of the FedEDAuth defense. It is assumed to be operated by a trusted, independent auditing authority analogous to existing third-party IC authentication bodies in the semiconductor industry and is inaccessible to any participating client or the central server. Its responsibilities are to maintain the golden reference dataset $D_{ref}$ (sourced from a certified IC testing authority or industry consortium), distribute the fixed feature extractor $g(\cdot)$ to all clients, receive their embedding-label pairs, score each client's embedding distribution against $D_{ref}$ via multi-metric anomaly detection, and issue verification tags to clients identified as authentic.

### 5.2.3 Clients

Each client maintains a private local dataset and participates in two sequential stages per training round:

- Authentication Stage: The client uses the distributed feature extractor $g(\cdot)$ to generate embeddings from its local dataset and submits the resulting embedding-label pairs to the authentication server.
- Training Stage: Only clients that receive a verification tag from the authentication server proceed to train their local ResNet-18 model and submit model updates to the central server. Clients that fail authentication are excluded from the current training round.

This two-server architecture ensures that the authentication process is fully isolated from the central training pipeline, protecting exchanged embeddings from reverse engineering while preventing poisoned gradients from ever reaching the central server.

## 5.3 Golden Reference Dataset: Assumptions, Acquisition, and Maintenance

A foundational assumption of FedEDAuth is a clean, trusted golden reference dataset $D_{ref}$ maintained by the authentication server. This is not a new assumption but standard practice in IC authentication, physical and electrical inspection (e.g., side-channel delay testing) already mandate certified reference parts and $D_{ref}$ could be curated by government-certified testing authorities, industry consortia (e.g., SEMI, IPC), or bootstrapped from supervised physical-inspection audits.

$D_{ref}$ is used strictly to establish the per-class descriptors ($\mu_c$, $\Sigma_c$, $\tau_c$), not to train a model. The current implementation uses only 500 images per class, with Ledoit-Wolf shrinkage keeping the 768×768 covariance well-conditioned at this size. As manufacturing standards evolve a stale $D_{ref}$ would misplace $\tau_c$ and inflate false positives, requiring a versioned update protocol that re-audits certified batches and redistributes refreshed descriptors. Detection also degrades when an honest client is highly heterogeneous relative to $D_{ref}$, where elevated Mahalanobis distances reflect genuine mismatch rather than poisoning which can be easily improved with the help of this framework.

## 5.4 Reference Embedding Distribution

The authentication server builds its trusted baseline from the golden reference dataset $D_{ref}$. Intuitively, the goal is to learn what authentic IC embeddings 'normally look like' for each class, where they sit in feature space and how they naturally spread so that a client whose embeddings deviate from this norm can be flagged. The reference images are first split 70%/30%. The 70% training split is passed through the frozen extractor $g(\cdot)$ to produce reference embeddings per class $c$, from which the server computes three descriptors:

- Class mean $\mu_c$: The center of the cloud of authentic embeddings for class $c$, where they sit on average.
- Covariance $\Sigma_c$: The shape and orientation of that cloud, i.e., the directions in which it naturally spreads, estimated with Ledoit-Wolf shrinkage. We keep the full 768×768 matrix rather than a diagonal approximation because poisoning triggers tend to push embeddings along low-variance, off-axis directions that a diagonal covariance would ignore.

- Mahalanobis threshold $\tau_c$: the radius of the 'normal' region, set to the 99th-percentile Mahalanobis distance among the reference embeddings. A client embedding farther than $\tau_c$ from the center is treated as an outlier.

The Mahalanobis distance of an embedding $x$,

$$d(x) = \sqrt{(x - \mu_c)^\top \Sigma_c^{-1} (x - \mu_c)}, \tag{5}$$

measures how far $x$ lies from the class center in units of the cloud's natural spread rather than in raw Euclidean distance. A point can be far in absolute terms yet still normal if it lies along a direction in which authentic embeddings naturally vary, and conversely a small off-axis displacement can be highly anomalous. The threshold therefore adapts to the true geometry of authentic embeddings instead of treating all directions equally.

The remaining 30% of $D_{ref}$ is held out as a calibration set, split into small groups that act as simulated honest clients. The server computes all three metrics on these groups and stores the scores and during scoring they anchor the "honest" baseline, so genuinely honest clients score near zero and only truly anomalous clients cross the exclusion threshold.

## 5.5 Client Embedding Submission

Each client $i$ applies the distributed feature extractor $g(\cdot)$ to its entire local dataset and transmits the resulting collection of embedding-label pairs to the authentication server:

$$E_i = \{(x_j^{(i)}, y_j^{(i)})\} \tag{6}$$

Critically, images are captured before any training-time augmentation or normalization, so the ViT's own preprocessing applies cleanly. For compromised clients the trigger overlay is applied before extraction, exactly as in their local training, so the authentication server sees a faithful representation of each client's true data distribution. No raw images or gradients are transmitted, only the final-layer [CLS] embeddings and their class labels.

## 5.6 Multi-Metric Anomaly Detection

FedEDAuth evaluates each client's submitted embeddings using three complementary metrics, each capturing a different way poisoned data betrays itself in feature space: individual embeddings straying outside the normal region (outlier fraction), the client's entire embedding cloud drifting off-center (mean shift), and the poisoned samples forming their own isolated pocket (micro-cluster).

### 5.6.1 Outlier Fraction ($F_i$)

This metric asks what fraction of a client's images land outside the normal authentic region. For each embedding $x$ in class $c$, the server computes its Mahalanobis distance $d(x)$ from $\mu_c$ using the full precision matrix $\Sigma_c^{-1}$, and flags the sample as an outlier if $d(x) > \tau_c$. The outlier fraction $F_i$ is the proportion of such outliers:

$$F_i = \frac{\text{number of outliers}}{\text{total samples}} \tag{7}$$

On its own, the raw outlier fraction is a comparatively weak signal for the fine-grained scratch triggers at 64×64 resolution, since few individual embeddings cross the 99th-percentile boundary. As the spread of outlier fractions among honest calibration clients is small, however, even a modest absolute increase produces a large robust-$z$ score once normalized is assigned the highest weight ($w_F$) among the three metrics.

### 5.6.2 Mean Shift ($M_i$)

This metric asks whether the client's entire embedding cloud has drifted off the authentic center, even when no single image is an outlier. For each class it measures the distance between the client's class mean $\hat{\boldsymbol{\mu}}_c^{(i)}$ and the reference mean $\boldsymbol{\mu}_c$, again in the whitened (cloud-shape-aware) space so that small shifts along low-variance directions are amplified:

$$M_i = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \sqrt{\max\left(0,\ (\hat{\boldsymbol{\mu}}_c^{(i)} - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}_c^{-1} (\hat{\boldsymbol{\mu}}_c^{(i)} - \boldsymbol{\mu}_c)\right)} \tag{8}$$

where $\hat{\boldsymbol{\mu}}_c^{(i)}$ is the mean of client $i$'s embeddings for class $c$. This is the most reliable detector for the scratch-style triggers used here: because the trigger is applied uniformly across all images regardless of class, it consistently shifts both class centroids in a common direction, producing a strong, reproducible signal even when individual sample displacements are modest.

### 5.6.3 Micro-Cluster Score ($C_i$)

This metric asks whether the poisoned samples form their own tight island separate from the authentic cloud. For each class, the client's embeddings are pooled with a subsample of reference embeddings and split by $k$-means ($k$=2). A cluster is flagged as suspicious when it is dominated by the client's own embeddings, its centroid sits far from the reference mean (in whitened space), and it forms a compact, isolated pocket. The score $C_i$ reflects the strength of such a client-dominated "trigger pocket", a tight sub-cluster that is absent from clean reference data.

### 5.6.4 Suspicion Scoring and Client Filtering

The three raw metric values are first normalized using a robust $z$-score. The normalization pool combines the live client scores with the stored calibration scores from the 30% reference holdout, so the "honest" baseline is always grounded in trusted data. This prevents the score distribution from shifting as the Byzantine fraction changes.

The final suspicion score for each client is:

$$S_i = w_F \cdot \max(0, \tilde{F}_i)^{\,p} + w_M \cdot \tilde{M}_i + w_C \cdot \tilde{C}_i \tag{9}$$

with default weights $w_F = 0.5$, $w_M = 0.3$, $w_C = 0.2$ and exponent $p = 2.0$, where $\tilde{F}_i$, $\tilde{M}_i$, $\tilde{C}_i$ denote the robust $z$-score normalized values. The outlier fraction

receives the highest weight because it measures direct distributional membership, while mean shift and micro-cluster scores provide complementary signals. Any client whose suspicion score exceeds the threshold $\kappa$ is excluded from the current training round.

The weights and threshold are configurable by operators. The feature extractor $g(\cdot)$ can also be swapped for a domain-specific encoder if needed, making *FedEDAuth* adaptable beyond IC authentication.

### 5.6.5 Workflow Summary

FedEDAuth operates in two phases. First, the authentication server builds the per-class reference descriptors ($\mu_c$, $\Sigma_c$, $\tau_c$) and calibration scores from the golden reference $D_{ref}$. Then, in *each training round*:

1. the authentication server distributes the frozen extractor $g(\cdot)$ to all clients;
2. each client embeds its local images and submits the [CLS] embedding–label pairs;
3. the server computes $F_i$, $M_i$, and $C_i$, normalizes them against the calibration baseline, and forms the suspicion score $S_i$;
4. clients with $S_i \leq \kappa$ receive a verification tag; the rest are excluded from the round;
5. verified clients train locally and send updates to the central server, which aggregates them with FedAvg and redistributes the global model.

Because authentication happens before any update is aggregated, no poisoned gradient ever reaches the global model, closing the blind spot that reactive aggregation defenses cannot.

# 6 Experimental Setup

## 6.1 Dataset and Hardware-Specific Augmentation

The experimental evaluation is conducted on a dataset comprising 6,387 authentic integrated circuit (IC) images [42], encompassing a diverse range of IC types including microcontrollers, memory chips, logic devices, and processors spanning multiple manufacturers and categories. To simulate the visual characteristics of counterfeit components, data augmentation is applied to generate 3,000 counterfeit IC images representing recycled, remarked, cloned, and black-topped categories. The final dataset therefore consists of 3,387 authentic and 3,000 counterfeit IC images.

More than ten overlay patterns emulates features associated with counterfeit ICs that include light-to-moderate scratches, smudges, fingerprint traces, textured surface irregularities, and gradual color fading that simulates aging, sometimes combined (e.g., faint scratches with smudges) while remaining subtle enough to resemble realistic tampering rather than overt damage. Additional variations in illumination, contrast, noise, and texture increase diversity. Sample augmented images appear in Figure 4. All images are kept at their native 64×64 resolution which was sufficient to preserve the visual cues required for counterfeit detection and the dataset is partitioned into training (70%), validation (15%), and testing (15%) splits.

The counterfeit class is generated through image augmentation rather than physically verified specimens, which risks the classifier partially learning augmentation

artifacts rather than genuine counterfeit indicators. To mitigate this, the augmentation pipeline uses different overlay types, illumination variations, and surface texture modifications than those used as attack triggers, ensuring a clean separation between the counterfeit generation and poisoning pipelines.

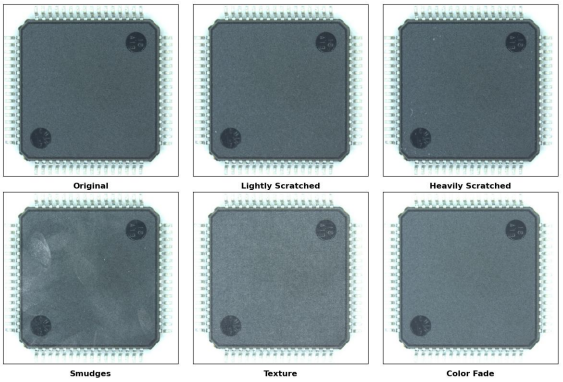



**Fig. 4** Synthetic counterfeit IC images produced by the augmentation pipeline.

## 6.2 Federated Learning Environment

The federated learning setup consists of 50 participating clients, reflecting a realistic deployment scenario involving multiple distributed supply chain entities. To further reflect real-world conditions, the training dataset is distributed randomly and unequally among all clients, introducing natural data heterogeneity across participants. Each experimental session consists of 10 training rounds. Within each round, all clients train a local ResNet-18 model on their private datasets for 50 epochs. Upon completion of the local training, model parameters are transmitted to the central server for aggregation into a new global model, which is then redistributed to all clients to initiate the subsequent training round.

ResNet-18 was selected for its well-established performance on image classification and its suitability for the 64×64 IC images used here. The unequal client data distribution is intentional, reflecting real-world deployments where supply chain entities possess varying quantities of IC images.

## 6.3 Byzantine Data Poisoning Attack Setup

To evaluate the vulnerability of standard FL pipelines, the Byzantine data poisoning attack is executed under three aggregation defenses (FedAvg, FedTrim, and Krum). As described there, 3 of 50 clients (collectively ≈4.46% of the training data) are compromised, with a different random three clients selected for each of the five seeds/experiments. Each configuration comprises two setups evaluated under the same defense:

- Setup 1 (Baseline): The FL system is trained for 10 rounds without poisoning. The best-performing model across all rounds is retained for evaluation.
- Setup 2 (Poisoned): The FL system is trained for 10 rounds with the Byzantine poisoning attack active. The best-performing model across all rounds is retained and compared against the baseline.

This paired evaluation design allows for a direct, controlled comparison of model performance with and without the poisoning attack under each defense mechanism.

## 6.4 FedEDAuth Defense Setup

To evaluate the effectiveness of FedEDAuth, the authentication framework is integrated into the federated learning pipeline as an independent pre-aggregation layer. In this configuration, three out of 50 clients are compromised using the Byzantine data poisoning attack, representing the more challenging attack scenario evaluated in this work.

At the beginning of each training round, the authentication server distributes the fixed feature extractor to all clients: a ViT-B/16 pretrained on ImageNet, used in frozen inference mode and producing 768-dimensional final-layer [CLS] embeddings. Input images are resized from 64×64 to 224×224 by bicubic interpolation and normalized with ImageNet channel statistics. All clients and the authentication server use identical preprocessing to ensure embedding consistency. ImageNet pretrained ViT features are well-suited as a fixed backbone for distribution-level anomaly detection because they yield semantically structured embedding spaces without domain-specific pretraining. Only authenticated clients proceed to train their local ResNet-18 for 50 epochs and submit parameters to the central server for FedAvg aggregation, repeated across all 10 rounds.

The golden reference dataset $D_{ref}$ maintained by the authentication server consists of clean, certified IC images representative of the authentic class distribution. In practice, this reference could be maintained by a trusted industry consortium, a government standards body, or a certified IC testing authority.

## 6.5 Baseline Aggregation Schemes and Evaluation Metrics

Three mainstream FL aggregation and defense schemes are evaluated as baselines to characterize both the attack's impact and FedEDAuth's effectiveness:

- Federated Averaging (FedAvg): Computes a weighted average of all client updates proportional to local dataset size, as defined in Equation 1.
- Trimmed Mean (FedTrim): Discards the most extreme gradient updates before averaging, as defined in Equation 2, with a trimming proportion of $\alpha = 0.2$.
- Krum: Selects the single client model most similar to its neighbors based on Euclidean distance, as defined in Equation 3 and 4.

Model performance is evaluated using five metrics that collectively characterize the ability of the global model to distinguish between authentic and counterfeit ICs:

- Accuracy: The proportion of correctly classified images across both classes.

- Precision: The proportion of images predicted as counterfeit that are truly counterfeit, measuring the model's false positive rate.
- Recall: The proportion of truly counterfeit images that are correctly identified, measuring the model's sensitivity to counterfeit detection.
- F1-Score: The harmonic mean of precision and recall, providing a balanced measure of classification performance across both classes.
- AUC-ROC: The area under the receiver operating characteristic curve, measuring the model's overall discriminative capability across all classification thresholds.

# 7 Results and Analysis

## 7.1 Counterfeit Detection Performance on Clean Data

Prior to evaluating the impact of adversarial attacks, the federated learning framework is assessed under non-poisoned conditions to establish a performance baseline. Utilizing the Federated Averaging (FedAvg) aggregation technique across 50 clients, 10 training rounds, and five independent random experiments, the global model achieves strong counterfeit IC detection performance, as summarized in Table 1. Results are reported as mean ± standard deviation across seeds.

**Table 1** Counterfeit IC Detection Model Performance Metrics

| **Metric** | **Mean ± Std** |
|---|---|
| Accuracy | 91.80% ± 0.76% |
| Precision | 87.96% ± 1.16% |
| Recall | 95.64% ± 0.54% |
| F1-Score | 91.64% ± 0.73% |
| AUC-ROC | 0.920 ± 0.007 |

The high recall of 95.64% ± 0.54% indicates that the model reliably identifies counterfeit images, minimizing false negatives, a critical property where undetected counterfeit components pose the greatest risk. Low standard deviations across seeds confirm that performance is stable across random client partitions and initializations, and the AUC-ROC of 0.920 ± 0.007 demonstrates strong discriminative capability. These results validate federated learning as a collaborative counterfeit IC detection paradigm that achieves high, reproducible performance without raw data exchange.

## 7.2 Impact of Byzantine Poisoning on Standard FL Pipelines

The proposed Byzantine data poisoning attack is evaluated against all three mainstream FL aggregation defense mechanisms. In each case, three compromised clients representing approximately 4.46% of the total training dataset inject subtle scratch-style triggers across their entire local datasets. Table 2 summarizes the performance of clean and poisoned models across all three defense configurations.

**Table 2** Comparison of Non-Poisoned and Poisoned Models Under Different Aggregation Methods

| Aggregation Method | Clean Accuracy | Poisoned Accuracy | Drop (%) |
|---|---|---|---|
| FedAvg | 91.80% ± 0.76% | 91.38% ± 0.80% | 0.42% |
| FedTrim | 91.13% ± 1.02% | 91.04% ± 1.02% | 0.09% |
| Krum | 82.84% ± 2.25% | 82.84% ± 2.25% | 0.00% |

1) Federated Averaging (FedAvg): Under FedAvg, the global model's accuracy decreased from 91.80% ± 0.76% to 91.38% ± 0.80% across 5 seeds, a mean drop of 0.42 percentage points. The false negative rate increased modestly under poisoning, indicating that the model has begun misclassifying counterfeit ICs as authentic. Because FedAvg averages all client updates and only a few are poisoned, their per-round influence is nearly negligible; however, it accumulates across rounds, gradually corrupting the global model's decision boundaries while remaining undetectable at the aggregation stage.

2) Trimmed Mean (FedTrim): Under FedTrim, accuracy decreased from 91.13% ± 1.02% to 91.04% ± 1.02%, a mean drop of 0.09 percentage points. FedTrim trims the most divergent model updates before averaging, making it effective against high-magnitude poisoning. However, because the proposed attack deliberately avoids producing large gradient deviations, the poisoned models are not identified as outliers and are not trimmed, allowing the attack to persist undetected across all training rounds.

3) Krum: Under Krum, accuracy was 82.84% ± 2.25% under both clean and poisoned conditions, with no measurable mean degradation. However, this apparent resistance to poisoning should not be interpreted as robustness as Krum's baseline accuracy is already substantially lower than FedAvg (82.84% vs 91.80%), reflecting its fundamental limitation of selecting only one client model per round and therefore failing to leverage the collaborative benefit of the full federated network. The absence of further degradation under poisoning is better understood as a floor effect than a defense success.

Crucially, this near-zero accuracy degradation (below 0.5 percentage points for all three defenses) does not diminish the threat but defines it. The attack operates below the detection threshold of standard performance monitoring, leaving no measurable accuracy signal while still corrupting the model's internal representations (Section 7.3), so operators have no performance-based indicator that the model has been compromised.

## 7.3 Grad-CAM Analysis of Model Corruption

To provide interpretable evidence of the attack's effect on the global model's internal representations, Gradient-weighted Class Activation Mapping (Grad-CAM) [43] is applied to compare feature focus between non-poisoned and poisoned models, as shown in Figure 5. In the non-poisoned model, activation heatmaps confirm that the model

concentrates attention on structurally meaningful IC regions, with no activation over the trigger area. In the poisoned model, activation migrates away from the structurally meaningful die center toward the trigger periphery across training rounds, with elevated activation emerging in the vicinity of the embedded scratch pattern, indicating that the global model has begun to treat the trigger region as a discriminative feature rather than a surface artifact.

This visualization provides two important insights beyond what accuracy metrics reveal. First, it confirms that the attack operates not merely as a statistical perturbation but as a targeted corruption of the model's feature learning process. The model is not simply noisier, it has learned the wrong features. Second, it explains why the accuracy drop is small while recall degrades more severely. The trigger association biases predictions toward the authentic class for triggered inputs, elevating false negatives without proportionally increasing false positives. The progressive shift of activation toward the trigger region was consistently observed across all five experimental seeds, confirming that embedding space corruption is a reproducible consequence of the attack rather than an artifact of any particular training run.

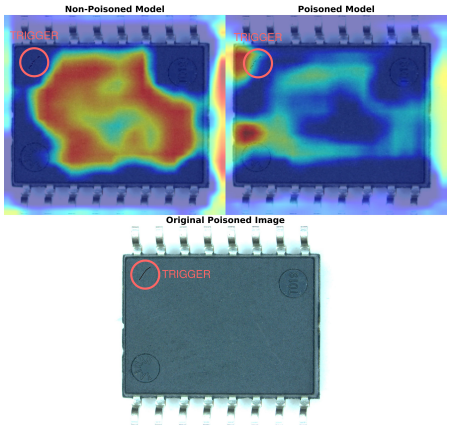



**Fig. 5** Heatmap generated by Grad-CAM showing areas the model focuses on in the non-poisoned vs poisoned models

## 7.4 FedEDAuth Authentication Performance

FedEDAuth is evaluated in the federated setting with 50 clients, of which 3 are compromised using the Byzantine data poisoning attack. The authentication server receives embedding-label pairs from all 50 clients and computes suspicion scores based on outlier fraction, class mean deviation, and micro-cluster behavior. The suspicion threshold is derived by introducing a held-out 30% calibration split of the golden reference and a bootstrap-based mean-shift normalization that sets honest client scores on an interpretable scale.

Across all 5 seeds, the three compromised clients ranked among the highest-scoring clients by a clear margin, with outlier fractions near 1.0 and substantially elevated

mean-shift and micro-cluster scores relative to honest clients, achieving a detection rate of 100% ± 0%. No poisoned client was missed in any seed.

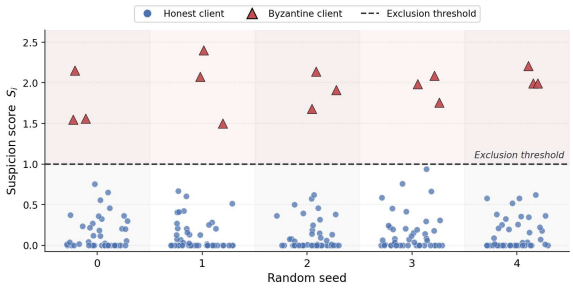



**Fig. 6** Per-client suspicion scores $S_i$ across five seeds

Table 3 presents an ablation across the three FedEDAuth metrics, evaluated individually over five seeds. AUC measures how cleanly a metric ranks poisoned clients above honest ones (0.5 = chance, 1.0 = perfect separation), while the Byz/Hon ratio is the mean score of poisoned clients divided by that of honest clients, basically how clearly the metric separates the malicious clients from the honest ones. Both $\tilde{F}$ and $\tilde{M}$ individually achieve AUC = 1.00 and 100% solo detection rate with 0 false positives. $\tilde{M}$ carries the largest margin (2.05× byz/honest ratio vs. 1.14× for $\tilde{F}$), making it the most robust to weaker or diluted triggers, as confirmed by the checkerboard experiment (7.6). $\tilde{C}$ alone reaches only 26.7% solo DR (AUC 0.782) but adds complementary coverage of client-dominated embedding pockets the mean-based metrics miss. The full three-metric detector achieves 100% DR at 0% FPR. Although a single raw embedding rarely crosses the per-sample outlier boundary at this resolution, the robust-$z$ normalization makes $\tilde{F}$ fully discriminative at the client level for high-magnitude anomalies such as the scratch trigger.

**Table 3** FedEDAuth Metric Ablation (Scratch Trigger, 5 Seeds)

| Metric | AUC | Byz/Hon Ratio | Solo DR @ FP=0 |
|---|---|---|---|
| $\tilde{F}$ — Outlier fraction | 1.000 | 1.14× | 100.0% |
| $\tilde{M}$ — Mean shift | 1.000 | **2.05**× | 100.0% |
| $\tilde{C}$ — Micro-cluster | 0.782 | 1.94× | 26.7% |
| Full detector ($\tilde{F}$+$\tilde{M}$+$\tilde{C}$) | — | — | 100.0% |

## 7.5 Comparison with Prior Federated Defenses

To test whether existing federated defenses detect the proposed attack, we implement three representative methods that operate on client model updates: FLTrust [35] (trust scoring against a clean root update), FLAME [36] (cosine-distance clustering),

and DeepSight [37] (output-divergence and update-energy inspection). The per-client updates are captured across 50 clients which train locally, scratch-poisoned for the Byzantine clients exactly as in the main experiments. FLTrust's root update is trained on the clean golden reference. Each defense assigns every client a suspicion score, which we assess two ways. The first is the AUC and the second is detection $k$ where we flag the $k$ most-suspicious clients and count how many are truly poisoned, fixing $k$=3 to the actual number of Byzantine clients. Using the same $k$ for every defense gives them an identical false-positive budget, so their detection rates are directly comparable, we do not tune $k$ per method, and the AUC which does not depend on $k$ confirms the comparison does not hinge on this choice. Table 4 reports both over the same five seeds.

**Table 4** Detection of the scratch poisoning attack by prior gradient-level defenses vs. FedEDAuth

| **Defense** | **AUC** | **DR** | **FPR** |
|---|---|---|---|
| FLTrust (root-update trust) | 0.58 | 6.7% | 6.0% |
| FLAME (update clustering) | 0.60 | 6.7% | 6.0% |
| DeepSight (behavior inspection) | 0.44 | 0.0% | 6.4% |
| **FedEDAuth** | **1.00** | **100.0%** | **0.0%** |

None of the three separates poisoned from honest clients. Their suspicion scores are statistically indistinguishable from random (AUC 0.44 - 0.60), and detection $k$ recovers Byzantine clients no better than the ≈6.7% expected by chance. Each Byzantine client trains honestly on triggered data rather than manipulating gradients, so its update carries no gradient-level signature. FLTrust assigns poisoned and honest clients near-identical trust scores, and FLAME finds no separable malicious cluster in any seed experiment. DeepSight scores even below chance (AUC 0.44) because the uniform trigger makes a poisoned client's data more homogeneous, yielding a less divergent update than honest clients produce. Update-level defenses are thus blind to a data-level attack that leaves the gradient signal intact, whereas FedEDAuth's pre-aggregation embedding analysis isolates every poisoned client (AUC 1.0) at zero false positives.

## 7.6 Trigger Generalization

To test whether detection is specific to scratch overlays, we evaluate a fundamentally different trigger: a 3×3 black-and-white checkerboard (9 pixels, 0.2% of the 64×64 frame) burned at a random corner. With detector parameters left entirely unchanged, the M*-fused detector again achieves detection rate of 100% at 0% false positive rate across all five seeds (Figure 7), with a tighter separation margin reflecting the checkerboard's smaller footprint. The cross-class consistency metric $\tilde{M}$ enables this generalization: any data-poisoning trigger displaces both class centroids, whereas honest variation perturbs at most one, so $\tilde{M}$ separates the two regardless of trigger geometry.

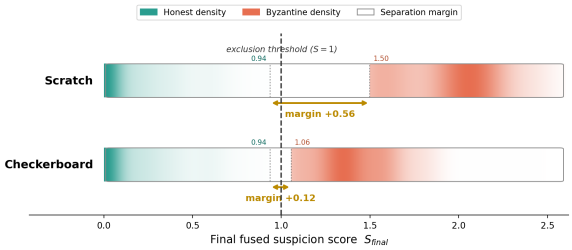



**Fig. 7** Honest and Byzantine suspicion-score densities for the scratch and checkerboard triggers.

## 7.7 Embedding-Space Visualization of the Byzantine Signature

Figure 8 shows where each client lands in the whitened mean-shift space the detector thresholds on. A client's two coordinates are how far its authentic and counterfeit class embeddings have drifted from the clean reference, in whitened (Mahalanobis) units. Honest clients form a dense cluster near the origin ($M \approx 5$ on both axes), while the Byzantine clients sit far out on *both* axes ($M_{auth} \approx 10$–$17$, $M_{cnt} \approx 8$–$10$; Byz/honest silhouette $= 0.788$, where 1 is perfect separation). They are extreme on both axes because the trigger is stamped on both classes and shifts both centroids together, whereas honest variation moves at most one the cross-class consistency property that the $M$ metric exploits.

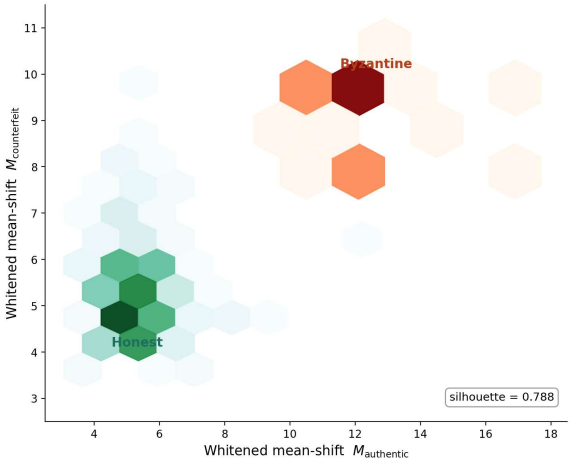



**Fig. 8** Hex-density of clients in the whitened mean-shift space.

## 7.8 Model Performance After FedEDAuth Filtering

Following the identification and exclusion of all three poisoned clients, the federated learning pipeline proceeds with only the 47 authenticated clients. The resulting global model achieves strong counterfeit IC detection performance, as summarized in Table 5.

**Table 5** Fedavg Model with FedEDAuth Performance Metrics

| Metric | Value |
|---|---|
| Accuracy | 90.94% ± 0.92% |
| Precision | 87.07% ± 1.14% |
| Recall | 94.80% ± 0.86% |
| F1-Score | 90.77% ± 0.92% |
| AUC-ROC | 0.91 ± 0.01 |

These results confirm that FedEDAuth identifies all compromised clients in every seed and restores near-baseline performance. Post-filtering accuracy (90.94%) and recall (94.80%) closely match the clean FedAvg baseline (91.80% and 95.64%). The preserved recall maintains the model's sensitivity to counterfeit detection, the most critical dimension in hardware security. The slight reduction is expected with a 0% false-positive rate all 47 honest clients are retained, but the three excluded Byzantine clients no longer contribute their data, marginally reducing training diversity.

### 7.9 Sensitivity Analysis: Poisoning Density

To evaluate the robustness of FedEDAuth under varying attack intensities, an additional experiment varies the proportion of poisoned images per compromised client across three levels: 90%, 70%, and 50%. The results are shown in Figure 9.

At 90% and 70% density all malicious clients are isolated (DR 100%); at 50% the poisoned distribution overlaps the threshold and a third of the Byzantine clients are missed (DR 66.7%), as a half-poisoned client's embeddings resemble a clean/poisoned mixture rather than a clearly anomalous one.

This edge case has limited consequence: at 50% density the signal is weak enough that FedAvg dilutes its influence. FedEDAuth's modular metrics and threshold can be extended with adaptive thresholding for lower densities and non-IID conditions, which we leave to future work.

## 8 Discussion & Hardware Security Implications

### 8.1 Federated Learning as a Hardware Security Enabler

The clean baseline results position FL not merely as a privacy-preserving alternative to centralized ML, but as an architecturally superior paradigm for long-term counterfeit IC detection. The global model adapts continuously to emerging counterfeiting strategies, scales with the diversity of participating entities, and removes the IP-exposure barrier that has deterred foundries from centralized programs. FL should thus be treated as infrastructure for hardware security intelligence, not merely a privacy tool.

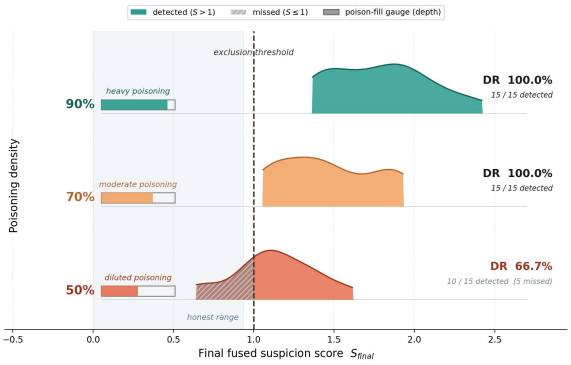



**Fig. 9** Distribution of Byzantine suspicion scores under decreasing poisoning density (15 Byzantine clients per level; 3 per seed × 5 seeds).

## 8.2 FedEDAuth as a Proactive Defense Paradigm

FedEDAuth's 100% malicious client detection rate across all experimental trials reflects a qualitative shift in the FL defense paradigm. Rather than filtering malicious contributions after aggregation, it intercepts compromised clients before any model update reaches the central server. By operating at the embedding level, it exploits a key asymmetry while subtle data poisoning yields gradient updates statistically similar to honest clients, the same poisoning produces a distinctive embedding-space signature (tight micro-clusters, systematic class-mean shifts, elevated Mahalanobis distances) that reflects the underlying data distribution rather than its gradient-level projection.

The authentication server's independence from both the central server and clients mirrors the role of third-party auditors in traditional IC authentication, aligning FedEDAuth with existing semiconductor verification workflows.

## 8.3 Adaptive Adversaries and the Software Pipeline Assumption

FedEDAuth's authentication guarantee rests on an implicit assumption that the client-side feature extractor $g(\cdot)$ and the local training pipeline operate over the same data. This closes a specific adaptive attack in which an adversary submits embeddings from a clean data pool while training locally on poisoned data passing authentication every round yet still corrupting the global model. The shared software pipeline prevents this decoupling.

A compromised foundry controlling its own infrastructure could, however, run a legitimate embedding pipeline separately from a modified training loop, submitting clean embeddings and poisoned updates independently. Enforcing this coupling requires hardware-rooted attestation (e.g., Intel SGX, AMD SEV, ARM TrustZone), through which the server can cryptographically verify that the unmodified pipeline is running a natural extension given that industrial semiconductor infrastructure commonly supports such primitives, but one left to future work.

A second adaptive strategy is an adversary who, aware of FedEDAuth's mean-shift detection, designs a trigger that minimizes class-centroid displacement while still corrupting the decision boundary. Such a trigger must simultaneously (a) shift the model's feature association toward the authentic class for triggered inputs and (b) keep client class means within the Mahalanobis threshold $\tau_c$ conflicting objectives that are difficult to satisfy with hardware-grounded triggers. Because effective IC-domain triggers must be visually coherent with surface features to evade human inspection, they carry correlated texture that shifts ViT embeddings systematically across the affected class. A trigger that shifts class means negligibly would have to be either imperceptibly small (weakening its effect) or spatially incoherent (visually detectable). Formally characterizing this efficacy–stealth tension, and quantifying the minimum trigger magnitude required to corrupt a ResNet-18 classifier while evading FedEDAuth, would yield a theoretical bound on the detection guarantee and is left to future work.

## 8.4 Broader Implications for Hardware Security

The findings extend beyond counterfeit IC detection. The same Byzantine data poisoning strategy could be adapted to corrupt FL models used for hardware trojan detection, side-channel analysis, fault injection monitoring, or supply chain integrity verification applications where the central server's inability to inspect client data creates the same fundamental vulnerability.

Conversely, FedEDAuth's embedding-based authentication is domain-agnostic. Its modular design, a substitutable feature extractor, configurable anomaly metrics, and tunable suspicion scoring adapts to any FL application, whether certified side-channel traces, authenticated trojan-free netlists, or verified fault-injection profiles. Authenticating clients through the statistical properties of their data representations, rather than their model-update parameters, is thus a generalizable strategy for securing federated intelligence across the IoT device fleets and hardware security pipelines. As FL becomes central to these workflows, standards such as AS6171 which today govern only physical and electrical inspection will need to evolve to specify trust-tier, golden-reference, and authentication-server requirements for ML-based collaborative detection pipelines.

## 8.5 Limitations

The counterfeit class is generated through image augmentation rather than physically verified specimens, so classifiers may partially learn augmentation artifacts rather than genuine counterfeit indicators. All experiments use a single IC image corpus with one family of scratch-style triggers; generalizability across IC types, imaging modalities, and trigger designs is not evaluated, and the 100% detection rate should be read within this scope.

Finally, the threat model does not empirically evaluate adaptive adversaries capable of decoupling their data pipelines. Hardware-rooted attestation is identified as the necessary countermeasure but remains unimplemented. Golden reference drift as manufacturing processes evolve is a further deployment challenge not captured here.

# 9 Conclusion

This work investigates federated learning as a privacy-preserving framework for collaborative counterfeit IC detection, addressing both its potential and its critical security vulnerabilities.

First, we demonstrated a novel Byzantine data poisoning attack against all mainstream FL defenses: FedAvg, Trimmed Mean, and Krum. With access to as little as 4.46% of the training data and no knowledge of the global model or other clients, the attack corrupts the global model's decision boundaries while producing mean accuracy degradations of 0.42 or less which is below the threshold of standard performance-based detection. Grad-CAM across all five random seed experiments confirms the impact is targeted corruption of internal features, not bulk accuracy loss. Because misclassifying even a small fraction of counterfeit ICs can be catastrophic for defense, medical, and critical-infrastructure systems, this stealthy corruption is an urgent, previously undercharacterized risk.

Second, to address this vulnerability, we propose Federated Embedding Distribution Authentication (FedEDAuth), a lightweight, proactive client authentication framework. A properly calibrated implementation achieves a 100% malicious client detection rate across all five seeds, with post-filtering performance of 90.94% $\pm$ 0.92% accuracy, 94.80% $\pm$ 0.86% recall, and an AUC-ROC of 0.91 $\pm$ 0.01.

Taken together, these findings underscore a principle for FL in hardware security. Decentralized collaboration is a powerful tool for counterfeit IC detection across a globalized supply chain, but its security cannot rest on reactive, gradient-level defenses alone. Proactive, data-aware authentication such as FedEDAuth is an essential complement, addressing the threat at its source rather than after the fact. Beyond counterfeit IC detection, the underlying authentication mechanism is a candidate defense for other FL-based hardware security applications, including IoT device authentication, hardware trojan detection, and side-channel analysis.

# Declarations


- Funding: The authors declare that no funds, grants, or other support were received.
- Competing interests: The authors have no competing interests.
- Consent for publication: All authors have reviewed the final version of the manuscript and consent to its publication
- Author contribution: Naseeruddin Lodge conducted the primary research and paper writing. Dr.Fareena Saqib and Dr. M. Yasin Akhtar Raja provided supervision and assisted with the writing, reviewing, and editing of the manuscript.


# References


[1] Akhoundov, D.: 2024 Annual Report. ERAI, Inc. https://www.erai.com/erai_blog/3187/_2024_annual_report Accessed 2026-06-08

[2] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data.

In: International Conference on Artificial Intelligence and Statistics (2016). https://api.semanticscholar.org/CorpusID:14955348

[3] Yang, Q., Huang, A., Fan, L., Chan, C.S., Lim, J.H., Ng, K.W., Ong, D.S., Li, B.: Federated learning with privacy-preserving and model ip-right-protection. Machine Intelligence Research **20**(1), 19–37 (2023) https://doi.org/10.1007/s11633-022-1343-2

[4] Chellapandi, V.P., Yuan, L., Brinton, C.G., Żak, S.H., Wang, Z.: Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. IEEE Transactions on Intelligent Vehicles **9**(1), 119–137 (2023)

[5] Aledhari, M., Razzak, R., Parizi, R.M., Saeed, F.: Federated learning: A survey on enabling technologies, protocols, and applications. IEEE access **8**, 140699–140725 (2020)

[6] Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., Jararweh, Y.: Federated learning review: Fundamentals, enabling technologies, and future applications. Information Processing & Management **59**(6), 103061 (2022) https://doi.org/10.1016/j.ipm.2022.103061

[7] Aramoon, O., Qu, G.: Impacts of machine learning on counterfeit ic detection and avoidance techniques. In: 2020 21st International Symposium on Quality Electronic Design (ISQED), pp. 352–357 (2020). IEEE

[8] Asadizanjani, N., Tehranipoor, M., Forte, D.: Counterfeit electronics detection using image processing and machine learning. In: Journal of Physics: Conference Series, vol. 787, p. 012023 (2017). IOP Publishing

[9] Ahmadi, B., Javidi, B., Shahbazmohamadi, S.: Automated detection of counterfeit ics using machine learning. Microelectronics Reliability **88**, 371–377 (2018)

[10] Bhure, C., Aklekar, D., Che, W., Saqib, F.: Vision transformers for counterfeit ic detection. In: 2024 IEEE Physical Assurance and Inspection of Electronics (PAINE), pp. 1–7 (2024). IEEE

[11] Xi, C., Varshney, N., Khan, M.S.M., Dalir, H., Asadizanjani, N.: Enhancing counterfeit detection of integrated circuits through machine learning-assisted thz-tds analysis. In: Terahertz, RF, Millimeter, and Submillimeter-Wave Technology and Applications XVII, vol. 12885, pp. 64–72 (2024). SPIE

[12] Saber Latibari, B., Nazari, N., Sasan, A., Homayoun, H., Satam, P., Salehi, S., Sayadi, H.: Transformers for secure hardware systems: Applications, challenges, and outlook. In: Proceedings of the Great Lakes Symposium on VLSI 2025, pp. 841–848 (2025)

[13] Stern, A., Botero, U., Shakya, B., Shen, H., Forte, D., Tehranipoor, M.: Emforced: Em-based fingerprinting framework for counterfeit detection with demonstration on remarked and cloned ics. In: 2018 IEEE International Test Conference (ITC), pp. 1–9 (2018). IEEE

[14] Gilmore, E.T., Frazier, P.D., Collins, I.J., Reid, W., Chouikha, M.: Infrared analysis for counterfeit electronic parts detection and supply chain validation. Environment Systems and Decisions **33**, 477–485 (2013)

[15] Nechiyil, A., Lee, R., Chapman, G.: A rapid, non-destructive method to detect counterfeit integrated circuits using a resonant cavity system. Instruments (2410-390X) **8**(3) (2024)

[16] Vakil, A., Niknia, F., Mirzaeian, A., Sasan, A., Karimi, N.: Learning assisted side channel delay test for detection of recycled ics. In: Proceedings of the 26th Asia and South Pacific Design Automation Conference, pp. 455–462 (2021)

[17] Surabhi, V.R., Krishnamurthy, P., Amrouch, H., Henkel, J., Karri, R., Khorrami, F.: Golden-free robust age estimation to triage recycled ics. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **42**(9), 2839–2851 (2023)

[18] Chuang, J., Lai, C., Griffith, D.: Counterfeit ic detection using rf excited signals and ai-assisted classification (2025)

[19] OpenSistemas: 8 Applications of Federated Learning Across the Globe. Accessed: 2025-02-14. https://opensistemas.com/en/8-applications-of-federated-learning/

[20] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv preprint arXiv:1811.03604 (2018)

[21] Zheng, Y., Chang, C.-H., Huang, S.-H., Chen, P.-Y., Picek, S.: An overview of trustworthy ai: advances in ip protection, privacy-preserving federated learning, security verification, and gai safety alignment. IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2024)

[22] Lee, G., Bowman, W., Dizon-Paradis, O., Dizon-Paradis, R., Wilson, R., Woodard, D., Forte, D.: Potentials and pitfalls of applying federated learning in hardware assurance. In: International Symposium for Testing and Failure Analysis, vol. 85212, pp. 302–313 (2025). ASM International

[23] Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W.: A survey on federated learning: challenges and applications. International journal of machine learning and cybernetics **14**(2), 513–535 (2023)

[24] Zhang, H., Liu, Y., He, X., Wu, J., Cong, T., Huang, X.: Sok: Benchmarking poisoning attacks and defenses in federated learning. arXiv preprint arXiv:2502.03801 (2025)

[25] Rosenfeld, E., Winston, E., Ravikumar, P., Kolter, Z.: Certified robustness to label-flipping attacks via randomized smoothing. In: International Conference on Machine Learning, pp. 8230–8241 (2020). PMLR

[26] Paudice, A., Muñoz-González, L., Lupu, E.C.: Label sanitization against label flipping poisoning attacks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 5–15 (2018). Springer

[27] Xie, C., Huang, K., Chen, P.-Y., Li, B.: Dba: Distributed backdoor attacks against federated learning. In: International Conference on Learning Representations (2019)

[28] Tomsett, R., Chan, K., Chakraborty, S.: Model poisoning attacks against distributed machine learning systems. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, vol. 11006, pp. 481–489 (2019). SPIE

[29] Fang, M., Cao, X., Jia, J., Gong, N.: Local model poisoning attacks to {Byzantine-Robust} federated learning. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 1605–1622 (2020)

[30] Xie, C., Koyejo, O., Gupta, I.: Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In: Uncertainty in Artificial Intelligence, pp. 261–270 (2020). PMLR

[31] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: Chiappa, S., Calandra, R. (eds.) Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 108, pp. 2938–2948. PMLR, ??? (2020). https://proceedings.mlr.press/v108/bagdasaryan20a.html

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

[33] Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: International Conference on Machine Learning, pp. 5650–5659 (2018). Pmlr

[34] Blanchard, P., El Mhamdi, E.M., Guerraoui, R., Stainer, J.: Machine learning with adversaries: Byzantine tolerant gradient descent. Advances in neural information processing systems **30** (2017)

[35] Cao, X., Fang, M., Liu, J., Gong, N.Z.: Fltrust: Byzantine-robust federated learning via trust bootstrapping. arXiv preprint arXiv:2012.13995 (2020)

[36] Nguyen, T.D., Rieger, P., Chen, H., Yalame, H., Möllering, H., Fereidooni, H., Marchal, S., Miettinen, M., Mirhoseini, A., Zeitouni, S., *et al.*: {FLAME}: Taming backdoors in federated learning. In: 31st USENIX Security Symposium (USENIX Security 22), pp. 1415–1432 (2022)

[37] Rieger, P., Nguyen, T.D., Miettinen, M., Sadeghi, A.-R.: Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. arXiv preprint arXiv:2201.00763 (2022)

[38] Lodge, N., Tambe, N., Aklekar, D., Saqib, F.: Counterfeit ic detection via federated learning: Exposure to byzantine data poisoning. In: 2025 IEEE Physical Assurance and Inspection of Electronics (PAINE), pp. 1–7 (2025). IEEE

[39] Lodge, N., Aklekar, D., Chadalavada, V., Tambe, N., Gholami, S., Alam, M., Saqib, F.: Fededauth–federated embedding distribution authentication for counterfeit ic detection. arXiv preprint arXiv:2605.15885 (2026)

[40] Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. Advances in neural information processing systems **32** (2019)

[41] Yin, H., Mallya, A., Vahdat, A., Alvarez, J.M., Kautz, J., Molchanov, P.: See through gradients: Image batch recovery via gradinversion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16337–16346 (2021)

[42] Reza, M.A., Crandall, D.J.: Ic-chipnet: Deep embedding learning for fine-grained retrieval, recognition, and verification of microelectronic images. In: 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), pp. 1–10 (2020). IEEE

[43] Gildenblat, J., contributors: PyTorch library for CAM methods. GitHub (2021)