Federated Defense in the Medical Edge: A Reputation-Aware Secure Aggregation Protocol for IoMT Intrusion Detection

Feras Fares AL-Mashakbah[1], Abdullah Alqammaz[2], Ala' Khalifeh[3]*,
Eman Fares Al Mashagbah[4], Essam Said Hanandeh[5]
Jerash University, Jerash, Jordan[1,2,5]
German Jordanian University, Amman, Jordan[3]
Philadelphia University, Amman, Jordan[4]

***Abstract*—Smart hospitals deploy Internet of Medical Things (IoMT) sensors and MQTT brokers to stream clinical telemetry over resource-constrained edge gateways. Centralized network intrusion detection systems (NIDS) expose sensitive traffic traces and create single-point failures; federated learning (FL) avoids raw-data centralization but remains vulnerable to client-side model poisoning and server-side inspection of client updates. This study presents Trust-Weighted Federated Aggregation (TW-Fed), a reputation-aware defense framework for federated learning that integrates 1) cosine similarity-based trust scoring using low-dimensional update sketches and 2) secure aggregation of trust- and data size-weighted model updates through pairwise masking with dropout recovery. TW-Fed has been evaluated on the CICIoMT2024 benchmark (Wi-Fi and MQTT subsets) with a four-class NIDS task (Benign, MQTT-Connect-Flood, TCP-DDoS, ARP-Spoofing) and 20 edge clients emulating IoMT gateways. In the clean setting, TW-Fed achieves 99.21% accuracy and 98.74% macro-F1, exceeding FedAvg by 0.82 macro-F1 points. Under label flipping with 30% malicious clients, TW-Fed sustains 97.06% accuracy versus 82.14% (FedAvg), 85.33% (FedProx), and 92.07% (Krum). Under backdoor injection, TW-Fed reduces attack success rate from 91.5% (FedAvg) to 6.8%. On Raspberry Pi 4 clients, TW-Fed adds 0.09 s local overhead per round while reducing server aggregation time by 34% relative to Krum. Secure aggregation increases uplink volume by 9.7% while preventing per-client update disclosure.**

***Keywords*—Federated learning; intrusion detection; internet of medical things; secure aggregation; poisoning attacks; reputation systems; MQTT security**

I. Introduction

IoMT deployments in smart hospitals couple bedside sensors, infusion pumps, and wearable monitors with edge gateways that forward telemetry to clinical systems over Wi-Fi and MQTT publish/subscribe fabrics [1], [2]. Network intrusion detection in this setting must operate under 1) strict privacy constraints on patient-adjacent metadata, 2) high availability requirements, and 3) device heterogeneity that induces non-IID traffic distributions across wards and clinical workflows [3]. Centralized NIDS pipelines collect packet traces at a core monitor; this design increases regulatory exposure and concentrates failure domains.

FL shifts model training to IoMT gateways by aggregating local updates rather than raw flows [4], [5]. Hospitals can thus train a global NIDS classifier without centralizing sensitive traces. However, FL introduces an integrity surface absent in centralized training: a compromised gateway can corrupt its local labels (label flipping) or embed a targeted backdoor into the shared model (backdoor injection) while remaining statistically similar on held-out benign traffic [6]–[8].

Beyond integrity, FL also creates *update confidentiality* concerns. If the server can inspect individual updates, it may perform update-inference or gradient inversion attacks that reconstruct sensitive properties of training data [9], [10]. Secure aggregation protects against an honest-but-curious server by ensuring it only learns an aggregate sum of client updates [11]. However, secure aggregation complicates robustness: many poisoning defenses rely on inspecting or filtering individual updates, which secure aggregation intentionally hides.

MQTT-based smart hospital segments amplify poisoning risk because broker-facing gateways often terminate device authentication and enforce topic policies; adversaries that obtain gateway control can mount both traffic-plane attacks (e.g., MQTT connect floods) and learning-plane attacks (poisoned updates) concurrently [12], [13]. To address this tension between robustness and confidentiality, TW-Fed separates (low-dimensional) *trust evidence* from (high-dimensional) masked updates.

The CICIoMT2024 is used as the primary benchmark because it captures multi-protocol IoMT traffic and explicitly includes Wi-Fi and MQTT attack categories under realistic device profiling [2], [14].The focus is on a four-class NIDS task aligned with common smart-hospital threat reports: Benign, MQTT-Connect-Flood, TCP-DDoS, and ARP-Spoofing.

A. Contributions

This study makes three key contributions:

- A threat-driven federated learning (FL) system model for IoMT-based network intrusion detection systems (NIDS) is developed to characterize label-flipping and backdoor injection attacks, with robustness evaluated under varying malicious-client ratios and non-IID data partitions.
- A reputation-aware aggregation rule, *TW-Fed*, is introduced based on cosine similarity-based trust scoring

*Corresponding author.

computed from low-dimensional update sketches, enabling the attenuation of low-trust client contributions without requiring prior knowledge of the proportion of malicious clients.

- Trust weighting is integrated with secure aggregation by masking *trust- and data-size-weighted* model updates and incorporating dropout recovery, thereby preserving the confidentiality of individual client updates while maintaining robustness and competitive latency on Raspberry Pi 4 gateways.

B. Design Goals

TW-Fed is guided by four deployment-oriented goals:

- Robustness under non-IID data: avoid assumptions that honest updates are tightly clustered.
- Update confidentiality: the server should not learn individual client updates in plaintext.
- Edge feasibility: additional per-round compute and communication must remain small for gateway-class devices.
- Graceful degradation: tolerate client dropouts, intermittent connectivity, and partial participation.

II. Related Work

A. FL for Healthcare and IoMT Security

Recent IoMT security surveys emphasize the need for lightweight, edge-capable IDS models and realistic datasets [3]. MQTT-specific IDS systems exist in centralized and federated forms; IDS-MA reports that FL can train MQTT-attack detectors without exporting raw traces, but it does not address poisoning robustness [13]. ARTEMIS and process-aware MQTT IDS frameworks demonstrate protocol-specific intrusion detection, yet they assume trusted training pipelines [15], [16].

B. Attacks on MQTT/IoMT Protocol Stacks

MQTT brokers and gateways admit flooding and malformed-packet attacks; protocol-aware defenses integrate parsing and strict validation into IDS layers [1], [12]. MQTT 5.0 also supports traffic patterns that adversaries can exploit for covert channels, complicating anomaly baselines [17]. CICIoMT2024 operationalizes these threats by executing multiple attack categories (including MQTT and spoofing) in a multi-device IoMT testbed [2].

C. FL Poisoning and Defenses

Model poisoning and data poisoning attacks degrade FL via malicious local updates [8], [18]. Backdoor attacks in FL induce targeted misclassification while preserving aggregate accuracy, making detection nontrivial [7], [19], [20]. Byzantine-robust aggregators (Krum, trimmed mean, coordinate-wise median, Bulyan) mitigate arbitrary updates but incur costs or assumptions that often break under heterogeneous, non-IID data [21]–[24]. Differential privacy reduces some forms of privacy leakage but does not directly prevent poisoning; DP noise can also reduce separability for IDS tasks under class imbalance [25]–[27]. Secure aggregation prevents server-side inspection of individual updates but interacts with robustness because anomaly detection often expects plaintext updates [9], [11]. TW-Fed resolves this tension by separating (low-dimensional) trust evidence from (high-dimensional) masked updates, and by applying trust- and size-weighting *before* masking.

III. System Model

A. Federated IoMT NIDS Training

An IoMT federated learning system is modeled with K gateways (clients), indexed by $k \in \{1, \dots, K\}$, and a coordinating server. Each gateway k observes local flow samples $\mathcal{D}_k = \{(\boldsymbol{x}_{k,i}, y_{k,i})\}_{i=1}^{n_k}$, where $\boldsymbol{x}_{k,i} \in \mathbb{R}^p$ denotes a feature vector extracted from Wi-Fi/MQTT/TCP/ARP flows and $y_{k,i} \in \mathcal{Y}$ is a class label. For the primary task, $\mathcal{Y} = \{\mathsf{Benign}, \mathsf{MQTT\text{-}CF}, \mathsf{TCP\text{-}DDoS}, \mathsf{ARP\text{-}S}\}$.

A global classifier $f(\boldsymbol{x}; \boldsymbol{w})$ with parameters $\boldsymbol{w} \in \mathbb{R}^d$ has been trained by minimizing:

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d} F(\boldsymbol{w}) &\triangleq \sum_{k=1}^{K} \frac{n_k}{n} F_k(\boldsymbol{w}), \\ F_k(\boldsymbol{w}) &\triangleq \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(f(\boldsymbol{x}_{k,i}; \boldsymbol{w}), y_{k,i}), \end{aligned} \tag{1}$$

with $n = \sum_k n_k$ and cross-entropy loss $\ell(\cdot, \cdot)$.

At round $t \in \{0, 1, \dots, T-1\}$, the server broadcasts $\boldsymbol{w}^{(t)}$ to a participating subset $\mathcal{S}^{(t)} \subseteq \{1, \dots, K\}$, $|\mathcal{S}^{(t)}| = m$. Each client runs E local epochs of SGD and returns an update:

$$\Delta \boldsymbol{w}_k^{(t)} \triangleq \boldsymbol{w}_k^{(t+1)} - \boldsymbol{w}^{(t)}. \tag{2}$$

Fig. 1 summarizes the considered smart-hospital deployment: IoMT devices generate traffic, gateways extract flow features and train locally, and the hospital server coordinates trust scoring and secure aggregation.

B. Threat Model: Label Flipping and Backdoor Injection

A subset $\mathcal{M}^{(t)} \subseteq \mathcal{S}^{(t)}$ of gateways is assumed to be compromised by an adversary. The adversary can modify local training data and the local training pipeline, while the server remains honest-but-curious (it follows the protocol but attempts to infer client updates) [11].

1) Label flipping: A malicious gateway applies a label corruption map $\phi : \mathcal{Y} \to \mathcal{Y}$ to a fraction $\rho \in (0, 1]$ of its samples:

$$y'_{k,i} = \phi(y_{k,i}), \quad \boldsymbol{x}'_{k,i} = \boldsymbol{x}_{k,i}. \tag{3}$$

The poisoning function ϕ is instantiated as targeted label flips to Benign for attack classes, reflecting a realistic poisoning strategy that degrades recall while preserving aggregate accuracy on dominant benign traffic [6].

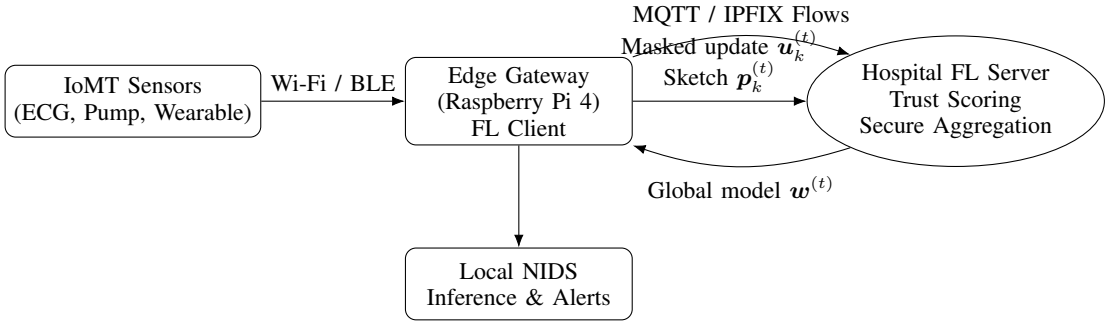

Fig. 1. Federated IoMT NIDS architecture: Sensors generate traffic; gateways train locally; the server scores cosine trust from low-dimensional sketches and aggregates masked trust-weighted updates.

2) Backdoor injection: A malicious gateway selects a trigger function $g : \mathbb{R}^p \to \mathbb{R}^p$ that modifies a small feature subset $\mathcal{I} \subset \{1, \ldots, p\}$ (e.g., setting normalized *Flow Duration* and *Packet Rate* to a narrow band). For a fraction $\beta \in (0, 1]$ of its local samples, it trains on

$$\boldsymbol{x}'_{k,i} = g(\boldsymbol{x}_{k,i}), \quad y'_{k,i} = y^\star, \tag{4}$$

where, $y^\star = \textsf{Benign}$ is the backdoor target. The attack success rate (ASR) is evaluated on triggered test samples:

$$\mathrm{ASR} = \Pr[f(g(\boldsymbol{x}); \boldsymbol{w}) = y^\star \mid y \neq y^\star]. \tag{5}$$

C. Notation Summary

Table I lists the main symbols used throughout the study.

TABLE I. MATHEMATICAL NOTATIONS

Symbol	Description
K	Number of clients (IoMT gateways).
$\mathcal{S}^{(t)}$	Participating client set at FL round t.
$\mathcal{M}^{(t)}$	Malicious client set at round t.
$\mathcal{D}_k$	Local dataset at client k, size n_k.
$\boldsymbol{x}_{k,i}, y_{k,i}$	Feature vector and label for sample i at client k.
$\boldsymbol{w}^{(t)} \in \mathbb{R}^d$	Global model parameters at round t.
$\Delta \boldsymbol{w}_k^{(t)}$	Local model update from client k at round t.
$\tau_k^{(t)} \in [0, 1]$	Trust score of client k (server-maintained reputation).
$\pi_k^{(t)}$	Aggregation weight for client k at round t.
$s_k^{(t)}$	Cosine similarity score used for trust update.
$\boldsymbol{R} \in \mathbb{R}^{q \times d}$	Public random projection matrix ($q \ll d$).
$\boldsymbol{p}_k^{(t)}$	Update sketch (projection) sent in the clear by client k.

IV. PROPOSED METHODOLOGY: TW-FED

TW-Fed combines 1) feature-driven local learning for IoMT flow classification, 2) a server-side cosine-trust detector computed from low-dimensional update sketches, and 3) a secure aggregation protocol that sums trust-weighted updates without exposing per-client updates.

A. Trust-Weighted Aggregation Rule

Normalized aggregation weights are defined as

$$\pi_k^{(t)} \triangleq \frac{\tau_k^{(t)} n_k}{\sum_{j \in \mathcal{S}^{(t)}} \tau_j^{(t)} n_j}, \qquad k \in \mathcal{S}^{(t)}, \tag{6}$$

And update the global model by

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \sum_{k \in \mathcal{S}^{(t)}} \pi_k^{(t)} \, \Delta \boldsymbol{w}_k^{(t)}. \tag{7}$$

TW-Fed reduces attacker influence by shrinking $\pi_k^{(t)}$ when $\tau_k^{(t)}$ decays. Unlike trimmed mean and Krum, TW-Fed does not require a known bound on $|\mathcal{M}^{(t)}|$ [21], [22].

B. Algorithm 1: Feature Extraction and Local Training

Each gateway parses CICIoMT2024 flow records (Wi-Fi and MQTT subsets) and constructs a feature vector. A fixed $p = 46$ feature set is used, including: *Flow Duration*, *Fwd/Bwd Packet Count*, *Packet Rate*, *Byte Rate*, *Inter-Arrival Time Mean*, *TCP Flags Counters*, and MQTT control-plane statistics (CONNECT/CONNACK ratio, PUBLISH rate, topic entropy).Features are standardized per-client using robust statistics (median and interquartile range) to reduce the impact of bursty IoMT traffic.

To keep Algorithm 1 compact (and avoid line overflows in IEEE columns), The mini-batch loss is denoted as:

$$L(\boldsymbol{w}; \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{(\boldsymbol{x}, y) \in \mathcal{B}} \ell(f(\boldsymbol{x}; \boldsymbol{w}), y). \tag{8}$$

C. Algorithm 2: Cosine-Trust Anomaly Detector

The server cannot robustly inspect masked updates under secure aggregation, so TW-Fed uses a low-dimensional sketch $\boldsymbol{p}_k^{(t)} = \boldsymbol{R} \Delta \boldsymbol{w}_k^{(t)}$ (with $q = 1024$) for trust scoring. A reference direction is computed as the coordinate-wise median of sketches:

$$\bar{\boldsymbol{p}}^{(t)} \triangleq \mathrm{med}\big(\{\boldsymbol{p}_k^{(t)} : k \in \mathcal{S}^{(t)}\}\big). \tag{9}$$

Algorithm 1 Feature Extraction & Local Training at Client k (CICIoMT2024 Wi-Fi/MQTT).

Input: Global parameters $\boldsymbol{w}^{(t)}$; local dataset $\mathcal{D}_k$; epochs E; batch size B; learning rate η.
Output: Local update $\Delta \boldsymbol{w}_k^{(t)}$; update sketch $\boldsymbol{p}_k^{(t)}$.
$\boldsymbol{w} \leftarrow \boldsymbol{w}^{(t)}$ **foreach** *flow record* $(\mathit{raw}_i, y_i) \in \mathcal{D}_k$ **do**
  Compute features $\boldsymbol{x}_i \in \mathbb{R}^{46}$ (FlowDur, PktRate, ByteRate, TCPFlags, MQTTConnectRate, ...) Robust standardization: $\boldsymbol{x}_i \leftarrow (\boldsymbol{x}_i - \text{median})/(\text{IQR} + \epsilon)$
end
for $e = 1$ **to** E **do**
  Shuffle $\mathcal{D}_k$ and form mini-batches of size B **foreach** *mini-batch* $\mathcal{B}$ **do**
    $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{w}} L(\boldsymbol{w}; \mathcal{B})$ $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta\, \boldsymbol{g}$
  end
end
$\Delta \boldsymbol{w}_k^{(t)} \leftarrow \boldsymbol{w} - \boldsymbol{w}^{(t)}$ $\boldsymbol{p}_k^{(t)} \leftarrow \boldsymbol{R}\, \Delta \boldsymbol{w}_k^{(t)}$ `// public random projection, q ≪ d`
return $(\Delta \boldsymbol{w}_k^{(t)}, \boldsymbol{p}_k^{(t)})$

Algorithm 2 Cosine-Trust Anomaly Detector at Server (Sketch-Based).

Input: Sketches $\{\boldsymbol{p}_k^{(t)}\}_{k \in \mathcal{S}^{(t)}}$; previous trust $\{\tau_k^{(t)}\}$; parameters $\lambda, a, \theta, \tau_{\min}$.
Output: Updated trust $\{\tau_k^{(t+1)}\}$; quarantine set $\mathcal{Q}^{(t)}$.
$\bar{\boldsymbol{p}}^{(t)} \leftarrow \text{med}\big(\{\boldsymbol{p}_k^{(t)}\}\big)$ $\mathcal{Q}^{(t)} \leftarrow \emptyset$ **foreach** $k \in \mathcal{S}^{(t)}$ **do**
  $s_k^{(t)} \leftarrow \frac{\langle \boldsymbol{p}_k^{(t)}, \bar{\boldsymbol{p}}^{(t)} \rangle}{\|\boldsymbol{p}_k^{(t)}\| \|\bar{\boldsymbol{p}}^{(t)}\| + \varepsilon}$ $\tau_k^{(t+1)} \leftarrow \text{clip}_{[0,1]}\Big(\lambda \tau_k^{(t)} + (1-\lambda)\sigma(a(s_k^{(t)} - \theta))\Big)$ **if** $\tau_k^{(t+1)} < \tau_{\min}$ **then**
    $\mathcal{Q}^{(t)} \leftarrow \mathcal{Q}^{(t)} \cup \{k\}$ `// quarantine (weight approaches 0)`
  end
end
return $(\{\tau_k^{(t+1)}\}, \mathcal{Q}^{(t)})$

Cosine similarity is then computed

$$s_k^{(t)} \triangleq \frac{\left\langle \boldsymbol{p}_k^{(t)}, \bar{\boldsymbol{p}}^{(t)} \right\rangle}{\left\| \boldsymbol{p}_k^{(t)} \right\| \, \left\| \bar{\boldsymbol{p}}^{(t)} \right\| + \varepsilon}. \tag{10}$$

Trust is updated with an exponential moving average and a soft threshold:

$$\tau_k^{(t+1)} \triangleq \text{clip}_{[0,1]}\Big(\lambda \tau_k^{(t)} + (1-\lambda)\, \sigma\Big(a\,(s_k^{(t)} - \theta)\Big)\Big), \tag{11}$$

where, $\sigma(u) = (1 + e^{-u})^{-1}$, $\lambda = 0.9$, $a = 12$, $\theta = 0.10$, and $\varepsilon = 10^{-12}$. Clients with trust values below $\tau_{\min} = 0.10$ are optionally quarantined, meaning that their weights approach zero instead of being completely removed.

D. Algorithm 3: Secure Aggregation of Trust- and Size-Weighted Updates

Secure aggregation is implemented via additive masking with pairwise seeds and dropout recovery following the standard construction [11]. To realize the weight $\pi_k^{(t)} \propto \tau_k^{(t)} n_k$ *without exposing individual updates*, both trust and data-size weighting are applied at the client before masking. (using the previous-round trust broadcast at the end of round $t-1$):

Algorithm 3 Reputation-Aware Secure Aggregation (TW-Fed-SA).

Input: Round t; participants $\mathcal{S}^{(t)}$; model $\boldsymbol{w}^{(t)}$; trust $\{\tau_k^{(t)}\}$; sizes $\{n_k\}$.
Output: Aggregated update $\Delta \boldsymbol{w}^{(t)}$ and new model $\boldsymbol{w}^{(t+1)}$.
Client k **(in parallel):** Compute $\Delta \boldsymbol{w}_k^{(t)}$ and sketch $\boldsymbol{p}_k^{(t)}$ (Alg. 1) Weight update: $\hat{\Delta} \boldsymbol{w}_k^{(t)} \leftarrow n_k \tau_k^{(t)} \Delta \boldsymbol{w}_k^{(t)}$ Generate pairwise masks $\{\boldsymbol{r}_{k,j}^{(t)}\}$ and secret shares for dropout recovery Send masked payload $\boldsymbol{u}_k^{(t)} = \hat{\Delta} \boldsymbol{w}_k^{(t)} + \sum_{j>k} \boldsymbol{r}_{k,j}^{(t)} - \sum_{j<k} \boldsymbol{r}_{j,k}^{(t)}$ and sketch $\boldsymbol{p}_k^{(t)}$
Server: Run Alg. 2 on $\{\boldsymbol{p}_k^{(t)}\}$ to compute $\{\tau_k^{(t+1)}\}$ for the next round Recover missing masks for dropped clients using secret shares Compute weighted sum $\boldsymbol{U}^{(t)} \leftarrow \sum_{k \in \mathcal{S}^{(t)}} \boldsymbol{u}_k^{(t)}$ Normalize and update: $\Delta \boldsymbol{w}^{(t)} \leftarrow \boldsymbol{U}^{(t)} / \sum_{k \in \mathcal{S}^{(t)}} n_k \tau_k^{(t)}$ $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} + \Delta \boldsymbol{w}^{(t)}$ Broadcast $\boldsymbol{w}^{(t+1)}$ and $\{\tau_k^{(t+1)}\}$

$$\hat{\Delta} \boldsymbol{w}_k^{(t)} \triangleq n_k\, \tau_k^{(t)}\, \Delta \boldsymbol{w}_k^{(t)}. \tag{12}$$

Client k forms a masked payload

$$\boldsymbol{u}_k^{(t)} \triangleq \hat{\Delta} \boldsymbol{w}_k^{(t)} + \sum_{j>k} \boldsymbol{r}_{k,j}^{(t)} - \sum_{j<k} \boldsymbol{r}_{j,k}^{(t)}, \tag{13}$$

where, $\boldsymbol{r}_{k,j}^{(t)} \in \mathbb{R}^d$ derives from a pairwise shared seed expanded by a PRG. The server receives $\{\boldsymbol{u}_k^{(t)}\}_{k \in \mathcal{S}^{(t)}}$ and obtains

$$\sum_{k \in \mathcal{S}^{(t)}} \boldsymbol{u}_k^{(t)} \;=\; \sum_{k \in \mathcal{S}^{(t)}} \hat{\Delta} \boldsymbol{w}_k^{(t)}, \tag{14}$$

after mask cancellation. When dropouts occur, clients secret-share their seeds so the server can reconstruct missing masks for dropped participants [11]. The server then normalizes by $\sum_{k \in \mathcal{S}^{(t)}} n_k \tau_k^{(t)}$ and applies the aggregated update.

E. Computational Complexity

Per round, trust scoring costs $O(mq)$ for sketch similarity plus the coordinate-wise median cost $O(mq)$ (for fixed q), while secure aggregation sums masked vectors in $O(md)$ time. Compared with distance-based robust selectors such as Krum, which require $O(m^2 d)$ pairwise distances, TW-Fed scales favorably with participation size m and remains compatible with secure aggregation.

V. Security and Privacy Analysis

A. Update Confidentiality Under an Honest-but-Curious Server

Secure aggregation ensures that an honest-but-curious server learns only an aggregate sum of masked client updates and cannot inspect any single client's update in plaintext [11]. This directly reduces exposure to update-inference attacks (e.g., gradient inversion) that attempt to reconstruct training examples from individual gradients [9] and complements

broader concerns about information leakage in collaborative learning [10].

B. What the Server Still Observes: The Sketch Channel

TW-Fed intentionally reveals a *low-dimensional* sketch $\boldsymbol{p}_k^{(t)} = \boldsymbol{R}\Delta\boldsymbol{w}_k^{(t)}$ to enable trust scoring. Random projection reduces dimensionality ($q \ll d$) and empirically provides adequate signal for anomaly detection, but sketches are not a cryptographic privacy guarantee. In deployments with strict privacy requirements, sketches can be further protected by 1) quantization, 2) norm clipping, and/or 3) adding DP noise at the sketch level. Differential privacy mechanisms provide formal bounds on leakage at the cost of accuracy [25]–[27].

C. Robustness Intuition and Practical Failure Modes

Let $\mathcal{H}^{(t)} = \mathcal{S}^{(t)} \setminus \mathcal{M}^{(t)}$ denote honest participants. The trust-weighted update can be decomposed as:

$$\sum_{k\in\mathcal{S}^{(t)}} \pi_k^{(t)}\Delta\boldsymbol{w}_k^{(t)} = \sum_{k\in\mathcal{H}^{(t)}} \pi_k^{(t)}\Delta\boldsymbol{w}_k^{(t)} + \sum_{k\in\mathcal{M}^{(t)}} \pi_k^{(t)}\Delta\boldsymbol{w}_k^{(t)}. \tag{15}$$

As trust scores for malicious clients decay, their total influence is upper-bounded by:

$$\sum_{k\in\mathcal{M}^{(t)}} \pi_k^{(t)} = \frac{\sum_{k\in\mathcal{M}^{(t)}} \tau_k^{(t)} n_k}{\sum_{j\in\mathcal{S}^{(t)}} \tau_j^{(t)} n_j}, \tag{16}$$

which approaches zero when malicious clients maintain consistently low cosine similarity to the median sketch direction.

TW-Fed is designed to degrade gracefully rather than hard-reject updates. Nevertheless, two practical failure modes remain relevant: 1) *adaptive attackers* that attempt to craft updates that both poison the model and mimic the sketch direction, and 2) *Sybil-style attacks* where many colluding clients submit correlated updates. Complementary defenses such as similarity-based Sybil mitigation (e.g., FoolsGold) [28] and stronger robust aggregation rules [23], [29] can be layered on top of TW-Fed when computational budgets allow. It is also emphasized that client-side weighting under secure aggregation is not directly verifiable by the server; accountable or verifiable variants of secure aggregation remain an important direction for future work.

VI. Experimental Setup

A. Dataset and Class Definition

The **CICIoMT2024** is used as the primary benchmark [2], [14]. The **Wi-Fi** and **MQTT** subsets are selected, and flows are filtered into four primary classes:

- Benign
- MQTT-Connect-Flood (MQTT-CF)
- TCP-DDoS
- ARP-Spoofing (ARP-S)

For Fig. 3, An auxiliary five-class ablation is also reported by adding Recon-PortScan, which CICIoMT2024 includes under reconnaissance-category attacks [2].

B. Client Topology and Non-IID Partition

A total of $K = 20$ IoMT gateways are emulated. Each client holds flows from a disjoint device group and a ward-specific traffic mix. Non-IID label proportions are generated using a Dirichlet distribution with concentration $\alpha \in \{0.1, 0.3, 1.0, 10.0\}$ over the class simplex (smaller α yields higher heterogeneity). A fixed number of $m = 16$ participating clients is used per round, and $\mathcal{S}^{(t)}$ is drawn uniformly without replacement.

C. Baselines

The following works are compared with the proposed work:

- FedAvg [4].
- FedProx with proximal coefficient $\mu = 0.01$ [30].
- Krum with $f = \lfloor 0.3m \rfloor$ assumed Byzantine bound in the attack experiments [21].

D. Model, Training, and Metrics

A lightweight MLP with two hidden layers (256 and 128 neurons) is used, GELU activations, and dropout 0.2. The parameter dimension is $d = 119{,}812$ (≈ 0.46 MB in FP32). Each round uses $E = 2$ local epochs, batch size $B = 256$, and Adam with learning rate $\eta = 10^{-3}$. Accuracy, Macro-F1, and class-wise false positive rate (FPR) are evaluated. For backdoor experiments, the attack success rate (ASR) on triggered samples is reported.

E. Key Hyperparameters

Table II summarizes the main training, trust, and secure aggregation parameters for reproducibility.

TABLE II. Key Hyperparameters and Protocol Settings (Default Unless Stated Otherwise)

Setting	Value
Clients / per-round participants	$K = 20$, $m = 16$
Rounds / local epochs / batch size	$T = 100$, $E = 2$, $B = 256$
Optimizer / learning rate	Adam, $\eta = 10^{-3}$
Model size	$d = 119{,}812$ params (MLP 256-128)
Sketch dimension	$q = 1024$
Trust update	$\lambda = 0.9$, $a = 12$, $\theta = 0.10$, $\tau_{\min} = 0.10$
Label flip setting	targeted to Benign, $\rho = 1$
Backdoor setting	target Benign, $\beta = 0.2$ (malicious clients)
Secure aggregation	pairwise masks + dropout recovery [11]

VII. Results and Discussion

A. Dataset Distribution

Table III summarizes the class distribution after filtering Wi-Fi and MQTT subsets. The benign majority induces a high-precision/low-recall failure mode under label flipping, which is explicitly tested.

B. Clean-Setting Detection Performance

Table IV reports accuracy and macro-F1 without adversaries. TW-Fed improves macro-F1 by stabilizing minority-class gradients under heterogeneous client distributions (trust weighting converges to near-uniform scores when no anomalies occur).

TABLE III. CICIoMT2024 Dataset Distribution for the Wi-Fi and MQTT Subsets (Filtered Four-Class Task)

Class	Train	Val	Test	Total
Benign	840,000	120,000	240,000	1,200,000
MQTT-Connect-Flood	105,000	15,000	30,000	150,000
TCP-DDoS	73,500	10,500	21,000	105,000
ARP-Spoofing	31,500	4,500	9,000	45,000
Total	1,050,000	150,000	300,000	1,500,000

TABLE IV. Detection Performance on CICIoMT2024 (Clean FL, $\alpha = 0.3$, $K = 20$, $m = 16$, $T = 100$)

Method	Accuracy (%)	Macro-F1 (%)
FedAvg	98.65	97.92
FedProx	98.84	98.05
Krum	98.91	98.11
TW-Fed (Ours)	**99.21**	**98.74**

C. Robustness to Label Flipping

Table V varies the malicious-client ratio $\gamma = |\mathcal{M}^{(t)}|/m$. Malicious clients flip attack labels to Benign with $\rho = 1$. FedAvg collapses under $\gamma \geq 0.3$ due to biased gradients. Krum remains robust at $\gamma = 0.3$ but degrades at $\gamma = 0.5$ under non-IID traffic because honest updates become dispersed and Krum can misidentify benign heterogeneity as adversarial outliers. TW-Fed sustains high accuracy by attenuating low-trust clients rather than hard selection.

Fig. 2 illustrates convergence behavior under $\gamma = 0.3$: FedAvg diverges, while TW-Fed converges faster than Krum due to reduced aggregation distortion and lower server-side filtering complexity.

To understand which attack categories degrade under poisoning, Fig. 3 reports per-class F1 for an auxiliary five-class ablation (including Recon-PortScan). TW-Fed preserves MQTT flood and reconnaissance detection under poisoning.

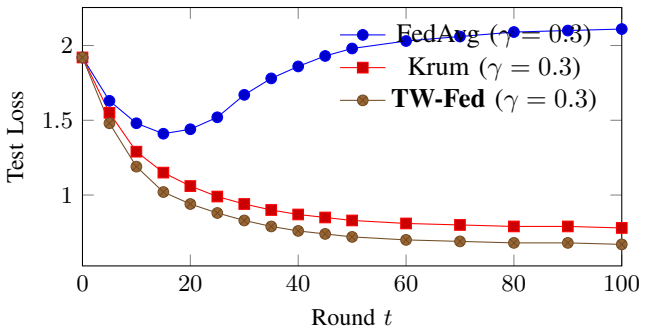

Fig. 2. Convergence under label flipping ($\gamma = 0.3$): FedAvg diverges; TW-Fed converges faster than Krum.

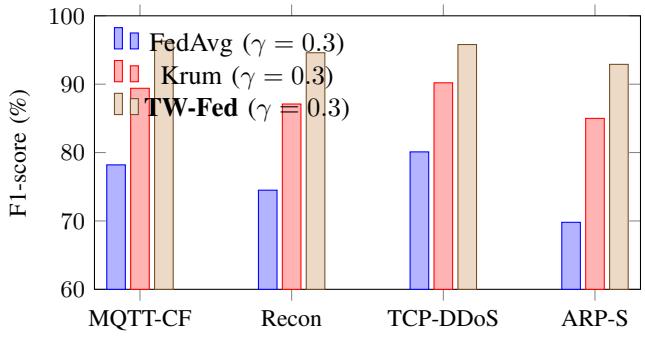

Fig. 3. Per-class F1 under $\gamma = 0.3$ label flipping (auxiliary five-class ablation includes Recon-PortScan).

D. Impact of Non-IID Data

Table VI reports performance across α. TW-Fed maintains macro-F1 under severe heterogeneity ($\alpha = 0.1$) because the trust update in (11) uses a soft score and does not assume IID geometry, unlike robust selectors that depend on distance concentration [24].

E. False Positive Rate by Attack Type

Table VII reports class-conditional FPR (probability of predicting a specific attack label on benign traffic). TW-Fed reduces ARP-Spoofing FPR, consistent with trust attenuation preventing poisoned updates from inflating rare-class decision regions.

F. Robustness to Client Dropouts

Secure aggregation protocols must tolerate partial participation and mid-round dropouts. Fig. 4 evaluates macro-F1 as a function of dropout probability p_d (the probability that a selected client fails to complete a round). TW-Fed-SA sustains higher macro-F1 because dropout recovery reconstructs missing masks and because trust weighting reduces reliance on any single client.

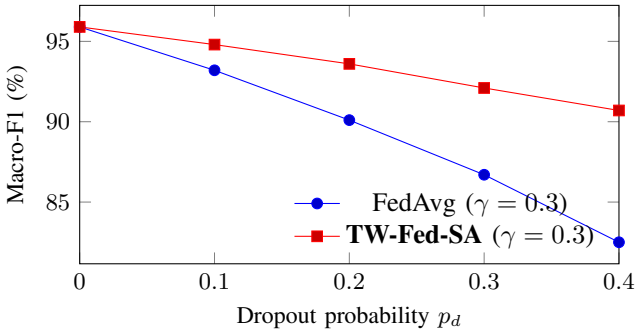

Fig. 4. Dropout robustness under label flipping ($\gamma = 0.3$). TW-Fed secure aggregation with dropout recovery sustains macro-F1 when up to 40% of selected clients fail mid-round.

G. Computational Latency and Communication Overhead

Table VIII reports per-round client compute time (mean over 100 rounds) and server aggregation time. Krum incurs

TABLE V. ROBUSTNESS UNDER LABEL FLIPPING (TARGETED FLIPS TO BENIGN, $\alpha = 0.3$, $T = 100$)

γ	FedAvg		FedProx		Krum		**TW-Fed (Ours)**	
	Acc (%)	Macro-F1 (%)	Acc (%)	Macro-F1 (%)	Acc (%)	Macro-F1 (%)	Acc (%)	Macro-F1 (%)
0.10	93.18	90.62	94.77	92.31	96.11	94.58	**98.12**	**97.01**
0.30	82.14	74.96	85.33	79.12	92.07	88.54	**97.06**	**95.88**
0.50	61.80	50.20	65.70	54.80	86.02	80.95	**93.41**	**90.27**

TABLE VI. IMPACT OF NON-IID PARTITIONS (DIRICHLET α; CLEAN SETTING)

α	Method	Accuracy (%)	Macro-F1 (%)
0.1	FedAvg	95.30	92.40
	FedProx	96.12	93.58
	Krum	95.85	93.10
	TW-Fed	**97.32**	**95.41**
0.3	FedAvg	98.65	97.92
	FedProx	98.84	98.05
	Krum	98.91	98.11
	TW-Fed	**99.21**	**98.74**
1.0	FedAvg	98.52	97.72
	FedProx	98.63	97.85
	Krum	98.71	97.95
	TW-Fed	**99.10**	**98.62**
10.0	FedAvg	98.83	98.14
	FedProx	98.91	98.21
	Krum	99.01	98.35
	TW-Fed	**99.23**	**98.78**

TABLE VII. FALSE POSITIVE RATE (FPR, %) ON BENIGN TEST FLOWS (CLEAN SETTING, $\alpha = 0.3$)

Method	FPR(MQTT-CF)	FPR(TCP-DDoS)	FPR(ARP-S)
FedAvg	1.42	1.05	1.90
FedProx	1.33	0.98	1.71
Krum	1.29	0.96	1.62
TW-Fed (Ours)	**0.88**	**0.73**	**0.70**

$O(m^2 d)$ pairwise distance costs. TW-Fed uses $O(md)$ masked summation plus $O(mq)$ sketch scoring, yielding lower server time while adding bounded client overhead for masking.

Fig. 5 reports total client-to-server uplink per round for $m = 16$ participants. TW-Fed-SA adds sketches and mask material, increasing volume by 9.7% relative to plaintext baselines.

H. Backdoor Injection Results

A feature trigger is injected on two standardized dimensions (Flow Duration and Packet Rate) and target Benign. With

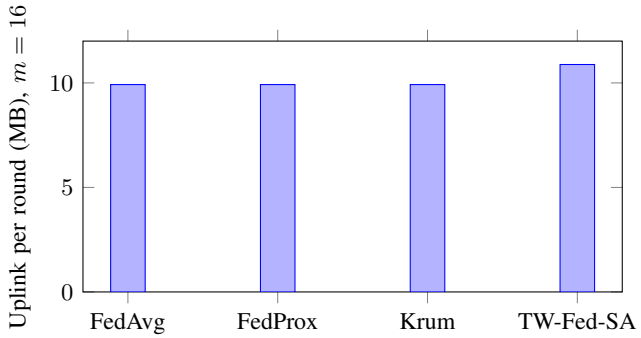

Fig. 5. Total client-to-server uplink per round for $m = 16$ participants. TW-Fed secure aggregation adds sketches and mask material, increasing volume by 9.7% relative to plaintext baselines.

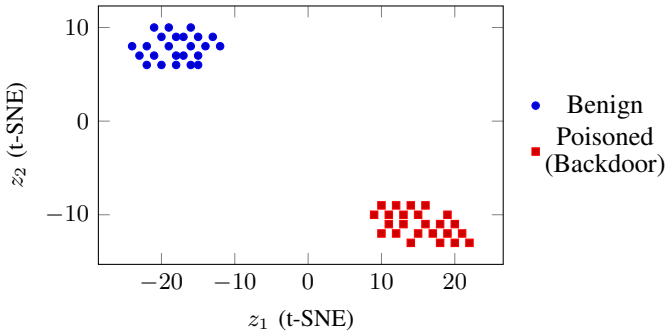

Fig. 6. Illustrative t-SNE embedding of penultimate-layer features of the global model. Backdoor-poisoned flows form a separable cluster; TW-Fed attenuates malicious influence, reducing ASR (Section VII-H).

$\gamma = 0.3$ malicious clients and $\beta = 0.2$ trigger fraction at malicious clients, FedAvg achieves ASR $= 91.5\%$ (attack persists after convergence). Krum reduces ASR to 21.3% but sacrifices macro-F1 by 4.9 points due to non-IID over-filtering. TW-Fed reduces ASR to 6.8% while maintaining macro-F1 $= 95.4\%$ on the four-class task. In the conducted runs, the trust trace shows that malicious clients typically converge to $\tau_k < 0.15$ within 12–18 rounds, after which their effective aggregation weight $\pi_k^{(t)}$ becomes negligible.

Fig. 6 provides an illustrative representation-space view (t-SNE on penultimate-layer embeddings) showing that backdoor-poisoned samples can form a separable cluster; TW-Fed reduces the persistence of this separation by attenuating malicious influence during training.

TABLE VIII. COMPUTATIONAL LATENCY (RASPBERRY PI 4 CLIENTS; $d = 119{,}812$; FP32; $m = 16$)

Method	Feature (s)	Train (s)	Mask (s)	Uplink (s)	Client Total (s)	Server Agg (ms)
FedAvg	0.18	2.74	0.00	0.21	3.13	62
FedProx	0.18	2.88	0.00	0.21	3.27	64
Krum	0.18	2.74	0.00	0.21	3.13	146
TW-Fed (Ours)	0.18	2.76	0.12	0.16	3.22	**96**

VIII. CONCLUSION

TW-Fed is presented as a reputation-aware secure aggregation protocol for IoMT intrusion detection in smart hospitals. TW-Fed computes cosine-based trust scores from update sketches and aggregates trust- and size-weighted masked updates with dropout recovery. On CICIoMT2024 (Wi-Fi and MQTT subsets), TW-Fed achieves 99.21% accuracy and 98.74% macro-F1 in the clean setting and maintains 97.06% accuracy under 30% malicious label flipping. Under backdoor injection, TW-Fed reduces ASR from 91.5% to 6.8% with bounded latency overhead on Raspberry Pi 4 gateways. As future work, a comprehensive study will be conducted by varying the sketch dimension (q) to assess its impact on trust-scoring fidelity and communication overhead.In addition, future work will evaluate TW-Fed using a broader range of classifier architectures, including CNN- and LSTM-based models, to determine whether its performance benefits remain consistent across different model types. Another important direction will be a more comprehensive security analysis under stronger adversarial settings, including malicious servers, server–client collusion, and compromised aggregation coordinators. Finally, future work will investigate the integration of verifiable trust evidence, such as protocol-compliant sketch commitments, together with accountable secure aggregation mechanisms that enable gateways to demonstrate the correctness of weighting and masking operations without exposing raw model updates.

REFERENCES

[1] A. Banks, E. Briggs, K. Borgendale, R. Gupta, and OASIS Message Queuing Telemetry Transport (MQTT) TC, "MQTT version 5.0," OASIS Standard, 2019, approved: 2019-03-07. http://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html.

[2] S. Dadkhah, E. C. Pinto Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. A. Ghorbani, "Ciciomt2024: A benchmark dataset for multi-protocol security assessment in IoMT," *Internet of Things*, vol. 28, p. 101351, 2024.

[3] M. L. Hernandez-Jaimes, A. Martinez-Cruz, K. A. Ramírez-Gutiérrez, and C. Feregrino-Uribe, "Artificial intelligence for IoMT security: A review of intrusion detection systems, attacks, datasets and cloud–fog–edge architectures," *Internet of Things*, vol. 23, p. 100887, 2023.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. PMLR, vol. 54, 2017, pp. 1273–1282.

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

[6] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *European Symposium on Research in Computer Security (ESORICS)*. Springer, 2020, pp. 480–501.

[7] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. PMLR, vol. 108, 2020, pp. 2938–2948.

[8] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 97, 2019, pp. 634–643.

[9] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[10] B. Hitaj, G. Ateniese, and F. Pérez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 603–618.

[11] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191.

[12] M. Husnain, K. Hayat, E. Cambiaso, U. U. Fayyaz, M. Mongelli, H. Akram, S. G. Abbas, and G. A. Shah, "Preventing MQTT vulnerabilities using IoT-enabled intrusion detection system," *Sensors*, vol. 22, no. 2, p. 567, 2022.

[13] A. Omotosho, Y. Qendah, and C. Hammer, "IDS-MA: Intrusion detection system for IoT MQTT attacks using centralized and federated learning," in *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2023, pp. 678–688.

[14] Canadian Institute for Cybersecurity (CIC), "CICIoMT2024 Dataset Page," https://www.unb.ca/cic/datasets/iomt-dataset-2024.html, 2024, accessed: 2026-02-06.

[15] E. Ciklabakkal, A. Dönmez, M. Erdemir, E. Suren, M. T. Yilmaz, and P. Angin, "ARTEMIS: An intrusion detection system for MQTT attacks in internet of things," in *2019 IEEE 38th Symposium on Reliable Distributed Systems (SRDS)*, 2019.

[16] P. Empl, F. Böhm, and G. Pernul, "Process-aware intrusion detection in MQTT networks," in *Proceedings of the Fourteenth ACM Conference on Data and Application Security and Privacy (CODASPY '24)*, 2024, pp. 91–102.

[17] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, and W. Mazurczyk, "Comprehensive analysis of MQTT 5.0 susceptibility to network covert channels," *Computers & Security*, vol. 104, p. 102207, 2021.

[18] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1605–1622.

[19] T. D. Nguyen *et al.*, "Backdoor attacks and defenses in federated learning," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 107166, 2024.

[20] S. Lu *et al.*, "Defense against backdoor attack in federated learning," *Computers & Security*, vol. 122, p. 102907, 2022.

[21] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[22] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 80, 2018, pp. 5650–5659.

[23] E.-M. El-Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in byzantium," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 80, 2018, pp. 3521–3530.

[24] M. Baruch, G. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[25] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conference (TCC)*, ser. Lecture Notes in Computer Science, vol. 3876. Springer, 2006, pp. 265–284.

[26] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[27] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2016, pp. 308–318.

[28] C. Fung, C. J. M. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018.

[29] K. Pillutla, S. M. Kakade, and Z. Harchaoui, "Robust aggregation for federated learning," *arXiv preprint arXiv:1912.13445*, 2019.

[30] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, 2020.