Dual-side Control for Coordinated Attack Detection in Federated Learning

Hanen Hamdani[1], Emna Benmohamed[1,2] and Hela Ltifi[1,3]

***Abstract*— Federated Learning (FL) enables collaborative model training while preserving data privacy. However, it remains highly vulnerable to poisoning attacks, particularly coordinated label- flipping attacks. In this paper, we propose DualFed, a dual-side defense framework for detecting coordinated poisoning behaviors in FL. DualFed combines client-side adaptive behavioral anomaly detection with a server-side Collective Behavioral Correlation (CBC) mechanism. On the client side, an Adaptive Adversar- ially Robust Statistics (AARS) mechanism integrates trimmed estimation with Exponential Moving Average (EMA) and Expo- nential Moving Variance (EMV) to robustly quantify abnormal performance degradation. On the server side, CBC aggregates client anomaly reports over a sliding temporal window to detect coordinated attacks. Once an attack is confirmed, DualFed ac- tivates a non-punitive self-recovery mechanism. Under non-IID settings, DualFed achieves benign accuracies exceeding 84.0MNIST, 52.0dataset, while reducing malicious attack accuracy to near zero and outperforming recent defense baselines.**

I. INTRODUCTION

Recent technological advances have significantly enhanced global connectivity and accelerated the growth of the Internet of Things (IoT). In particular, developments in Artificial Intelligence (AI) have enabled the emergence of Federated Learning(FL), introduced by McMahan et al. [1]. FL is a privacy-preserving paradigm that allows collaborative model training without centralizing data [2], [3]. It is particularly valuable in sensitive domains, where data sharing is prohibited. Despite its privacy advantages, FL introduces new security vulnerabilities. Its decentralized nature inherently expands the attack surface, as malicious participants can compromise the global model through data or model poisoning. Among the most dangerous threats are coordinated attacks, where multiple malicious clients collude to embed backdoors into the global model [4]. In such attacks, malicious clients synchronously manipulate their local training data. Consequently, traditional defenses based on client filtering or robust aggregation often fail to detect them. Several defenses have been proposed, such as Krum [5], Trimmed Mean [6], and client filtering [7], [8]. These methods typically rely on distance-based outlier detection or coordinate-wise robust statistics and often fail to effectively address coordinated poisoning attacks. This reinforces the growing need for intelligent, modular, and collaborative defense mechanisms capable of detecting coordinated threats without compromising privacy or performance. In this work, we propose a dual-sided defense mechanism named DualFed. Designed as a modular defense against coordinated label-flipping attacks, DualFed combines client-side anomaly monitoring with server-side temporal correlation. Upon detection, DualFed triggers a silent self-recovery by resetting the final fully connected layer using Kaiming initialization [9]. The backdoor is then effectively erased while preserving the learned features in the earlier layers. This correction is privacy-preserving and fully compatible with standard FL frameworks. Importantly, DualFed is designed as a plug-in defense: It can be seamlessly integrated into existing FL systems and further combined with complementary security modules, such as robust aggregation or client filtering. Consequently, DualFed is not only a standalone solution but also a complementary security layer that enhances the resilience of FL environments against sophisticated coordinated threats. The main contributions of this work are: (1) a novel client-side adaptive anomaly detection mechanism based on robust statistical modeling, (2) a server-side Collective Behavioral Correlation (CBC) mechanism for detecting coordinated poisoning attacks, (3) a non-punitive self-recovery strategy. Finally, we evaluate DualFed on MNIST [10], CIFAR-10 [11], and the Stroke Prediction dataset [12] under both IID and non-IID settings. Following this introduction, Section 2 reviews related work. Section 3 presents DualFed, detailing its architecture. Section 4 describes the experimental setup and provides a comprehensive evaluation of DualFed's performance. Finally, Section 5 concludes the paper.

II. RELATED WORK

A comprehensive review of the literature confirms that poisoning attacks in FL are most commonly categorized into data poisoning and model poisoning [13], [14]. In data poisoning, adversaries compromise the integrity of local training datasets, with label flipping being a prominent example [15], [16]. Conversely, model poisoning attacks directly manipulate the transmitted model updates through gradient manipulation [17], model replacement [18], or adaptive poisoning [19]. A more refined classification of attacks based on their operational strategy has recently been introduced in [20], [21], [22] .

Under this taxonomy, attacks are divided into two main categories: isolated and coordinated. In isolated attacks, malicious clients act independently. In coordinated attacks, multiple malicious clients synchronize their updates to amplify the attack's impact. Recent studies have shown that such coordinated attacks are particularly dangerous [23].

[1]Hanen Hamdani, Emna Benmohamed and Hela Ltifi are with Research Groups in Intelligent Machines University of Sfax,BP 1173, 3038 ,Tunisia hanen.hamdani@gmail.com

[2] emna.benmohamed@enis.tn

[3] hela.ltifi@ieee.org

The evolution of attacks in IoT systems calls for more sophisticated defense mechanisms. Consequently, developing such defenses has become a challenging research topic in the field of cybersecurity.

Statistical defense methods detect poisoning by analyzing the geometric properties of model updates. For example, Krum [5] is among the earliest and most cited defenses. It selects the gradient closest to the majority based on Euclidean distances. Despite being resilient to Byzantine failures, Krum fails to defend against coordinated attacks . DWAMA [24] uses Mahalanobis distance to flag suspicious updates and adjusts their aggregation weights proportionally to their anomaly scores. Suspicious clients are not excluded from the federation. FLPD [25] combines multiple similarity metrics (cosine, Euclidean) to identify malicious clients and correct gradients at the server side. MCDFL [26] detects label-flipping by assessing local data quality via latent space reconstruction. However, evaluating clients independently makes it ineffective. CBDF [27] detects coordinated attacks by identifying excessive similarity among client updates. Instead of flagging outliers, it monitors the alignment of model gradients. It exploits the fact that synchronized updates from malicious clients are statistically atypical in non-IID FL. This makes CBDF uniquely suited to detect collusion without relying on robust aggregation.

Beyond statistical methods, a second class of defenses is defined in the literature. It is inspired by biological systems. In [28] an immune-inspired framework for threat detection in FL is proposed. It uses an adaptive learning memory to control attacks in real time. Similarly, [29] models B-cell and T-cell dynamics by using a Reinforcement Learning (RL) agent to detect and isolate malicious nodes. FL-WBC [30] introduces a client-side defense based on the Attack Effect on Parameter (AEP) estimator. This approach treats FL clients as active defenders, not just data sources. In [31], a Genetic Algorithm (GA) is used to optimize a data quality threshold at each client to filter out poisoned data before training.

Methods such as Krum and DWAMA exclusively rely on gradient analysis and neglect client-side behavioral feedback. Client-based defenses like FL-WBC and MCDFL treat federated clients in isolation. Consequently, temporal and cross-client correlations are overlooked. A common limitation of existing methods lies in their reliance on update exclusion strategies. This exclusion is exploited by poisoning clients to amplify the apparent deviation of honest non-IID clients and ultimately push them out of the aggregation process. DualFed addresses these limitations by redefining clients as active security sensors rather than passive data holders. It combines local monitoring, temporal correlation of clients' anomaly reports, and silent model self-recovery.

III. Proposed Approach: DualFed

A. Proposed Architecture

To detect and mitigate coordinated label-flipping attacks in FL, DualFed operates through a dual-side monitoring and recovery framework, as illustrated in Fig. 1.

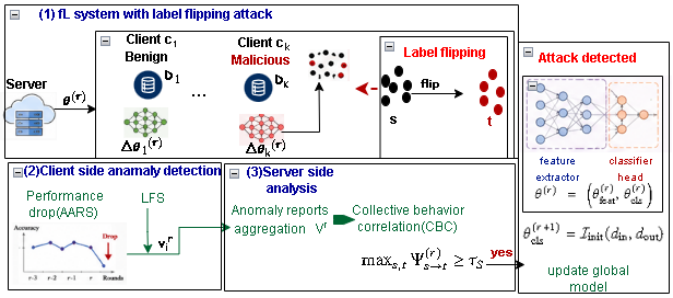

Fig. 1. DualFed architecture for coordinated label-flipping attack detection.

At each communication round, clients receive the global model $\theta^{(r)}$ and perform local training on their private datasets. To monitor local behavior, each client computes two complementary indicators: (i) a performance drop (PD) confidence score $\rho_i^{(r)}$, which measures abnormal degradation in local accuracy, and (ii) a label-flip suspicion (LFS) score $\xi_i^{(r)}(s \to t)$, extracted from the local confusion matrix to capture abnormal class-transition patterns. The PD statistics are processed using the proposed Adaptive Adversarially Robust Statistics (AARS) module to avoid reliance on pre-calibrated benign statistics. Each client then transmits a compact anomaly report $\mathbf{v}_i^{(r)} \in [0,1]^2$ to the server. Upon receiving these reports, the server performs a robust weighted aggregation and computes the CBC metric over a sliding temporal window of W rounds. CBC effectively captures synchronized behavioral anomalies across clients to identify coordinated collusion. Once the CBC score exceeds the detection threshold τ_S, DualFed triggers a silent, non-punitive self-recovery mechanism. This mechanism preserves the learned feature extractor while performing a hard reinitialization of the classifier head.

B. Federated Learning Setting and Notation

We consider an FL system composed of a finite set of N clients $\mathcal{C} = \{c_1, \ldots, c_N\}$. The objective is to collaboratively train a shared model $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by $\theta \in \mathbb{R}^d$ Each client c_i owns a local dataset $\mathcal{D}_i = \{(\mathbf{x}, y)\}$ and minimizes:

$$F_i(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_i}\big[\ell(f_\theta(\mathbf{x}), y)\big], \tag{1}$$

where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ denotes a differentiable loss function (e.g., cross-entropy). The global optimization objective is:

$$\min_\theta F(\theta) = \sum_{i=1}^{N} p_i F_i(\theta), \qquad p_i = \frac{|\mathcal{D}_i|}{\sum_{j=1}^{N} |\mathcal{D}_j|} \tag{2}$$

Training proceeds over communication rounds $r \in \mathbb{N}$. At each round, the server selects a subset of clients $\mathcal{S}_r \subseteq \mathcal{C}$, broadcasts the current global model $\theta^{(r)}$, and receives local model updates $\Delta\theta_i^{(r)}$:

$$\theta^{(r+1)} = \theta^{(r)} + \sum_{i\in\mathcal{S}_r} p_i \Delta\theta_i^{(r)} \tag{3}$$

C. Targeted Label-Flipping Operator

We consider a targeted label-flipping attack defined by a source label $s \in \mathcal{Y}$ and a target label $t \in \mathcal{Y}$, with $s \neq t$:

$$\phi_{s\to t}(y) = \begin{cases} t, & y = s \\ y, & \text{otherwise} \end{cases} \tag{4}$$

For a malicious client c_k, the poisoned local dataset becomes:

$$\mathcal{D}_k^{\phi} = \{(\mathbf{x}, \phi_{s\to t}(y)) : (\mathbf{x}, y) \in \mathcal{D}_k\} \tag{5}$$

The malicious local objective associated with the client c_k deviates from the benign objective in Eq. (2) and becomes:

$$F_k^{\phi}(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k}\left[\ell(f_\theta(\mathbf{x}), \phi_{s\to t}(y))\right] \tag{6}$$

At a given round r, client c_k returns the corresponding poisoned model update, denoted by $\Delta\theta_k^{(r)}(\phi)$.

D. Synchronized Collusion Dynamics

Let $\mathcal{M} \subset \mathcal{C}$ denote the set of malicious clients. A coordinated attack emerges when adversarial clients apply identical label-flipping strategies, producing strongly aligned model updates:

$$\forall\, c_k, c_{k'} \in \mathcal{M} \cap \mathcal{S}_r : \quad \cos\big(\Delta\theta_{c_k}^{(r)}(\phi), \Delta\theta_{c_{k'}}^{(r)}(\phi)\big) \geq 1 - \varepsilon, \tag{7}$$

where $\varepsilon \in (0, 1)$ is a tolerance parameter and $\cos(\cdot,\cdot)$ denotes cosine similarity, which measures the directional alignment between the poisoned updates of malicious clients c_k and $c_{k'}$ at round r. Let p_k denote the aggregation weight of client c_k defined in Eq. (2), the aggregated malicious drift injected into the global model at round r is

$$\Delta\theta_{\mathcal{M}}^{(r)} = \sum_{c_k \in \mathcal{M}\cap\mathcal{S}_r} p_k\, \Delta\theta_k^{(r)}(\phi) \tag{8}$$

DualFed deploys a dual-side control mechanism operating jointly at both the client and server levels.

E. Client-Side Behavioral Monitoring

At each communication round r, every client computes two complementary behavioral indicators: a PD score $\rho_i^{(r)}$ and an LFS score $\xi_i^{(r)}$.

*1) **Performance Drop Quantification**:* The PD metric corresponds to a sudden degradation of local model accuracy between consecutive rounds. Formally, the local accuracy of client c_i having a private validation dataset $\mathcal{D}_i^{\text{eval}}$ is defined as:

$$A_i^{(r)} = \frac{1}{|\mathcal{D}_i^{\text{eval}}|} \sum_{(\mathbf{x},y)\in\mathcal{D}_i^{\text{eval}}} \mathbf{1}(f_{\theta^{(r)}}(\mathbf{x}) = y) \tag{9}$$

The immediate PD is then:

$$\delta_i^{(r)} = A_i^{(r-1)} - A_i^{(r)} \tag{10}$$

A large positive value of $\delta_i^{(r)}$ indicates abnormal degradation of local performance.

*2) **Adversarially Robust Statistics**:* To quantify the statistical significance of this degradation, each client applies the AARS mechanism to eliminate the dependency on pre-calibrated benign statistics. The AARS framework is based on robust trimmed estimation theory [32], [33]. It operates during the learning process from the first communication round, even under adversarial contamination. We define the contamination rate $\alpha_r \in [0, 1)$ as $\alpha_r = \frac{|\mathcal{M}\cap\mathcal{S}_r|}{|\mathcal{S}_r|}$. Under adversarial contamination, the observed performance drop is modeled as:

$$\delta_i^{(r)} = (1 - \alpha_r)\, \delta_i^{\text{benign}} + \alpha_r\, \delta_i^{\text{adv}} \tag{11}$$

where δ_i^{benign} is the PD expected under clean federated aggregation, and δ_i^{adv} represents degradation caused by adversarial contamination. To quantify the statistical significance of the observed PD, each client normalizes $\delta_i^{(r)}$ using a z-score with adaptive mean and variance. Let K denote the window size; to mitigate the influence of adversarial outliers, the client c_i maintains a sliding window:

$$\mathcal{W}_i^{(r)} = \{\delta_i^{(r-K+1)}, \ldots, \delta_i^{(r)}\} \tag{12}$$

A trimmed estimator is first computed over the sliding window $\mathcal{W}_i^{(r)}$ to reduce the influence of adversarial outliers:

$$\tilde{\delta}_i^{(r)} = \text{Trim}_\gamma\left(\mathcal{W}_i^{(r)}\right), \tag{13}$$

where Trim_γ removes the $\lfloor \gamma K \rfloor$ largest and smallest observations before averaging the remaining samples. To maintain a stable long-term behavioral reference, Exponential Moving Average (EMA) and Exponential Moving Variance (EMV) recursively accumulate the client's historical statistics as:

$$\mu_{\delta,i}^{(r)} = (1 - \beta)\, \mu_{\delta,i}^{(r-1)} + \beta\, \tilde{\delta}_i^{(r)}, \tag{14}$$

$$\sigma_{\delta,i}^{2(r)} = (1 - \beta)\, \sigma_{\delta,i}^{2(r-1)} + \beta \left(\tilde{\delta}_i^{(r)} - \mu_{\delta,i}^{(r-1)}\right)^2, \tag{15}$$

where $\beta = \frac{2}{K+1}$ is used to align the EMA memory horizon with the sliding-window size K. The anomaly score is:

$$z_i^{(r)} = \frac{\delta_i^{(r)} - \mu_{\delta,i}^{(r)}}{\sigma_{\delta,i}^{(r)}} \tag{16}$$

Finally, the suspicion confidence score is defined using a sigmoid mapping:

$$\rho_i^{(r)} = \frac{1}{1 + e^{-z_i^{(r)}}} \tag{17}$$

*3) **Client-Side Label-Flip Suspicion**:* To capture abnormal class-targeted prediction behaviors, each client computes a local confusion matrix $C_i^{(r)}$, where $C_i^{(r)}[s, t]$ counts samples whose true label is s and predicted label is t.

The LFS score is defined as:

$$\xi_i^{(r)}(s \to t) = \frac{C_i^{(r)}[s, t]}{\sum_{j=1}^{|\mathcal{Y}|} C_i^{(r)}[s, j]} \tag{18}$$

Large values of $\xi_i^{(r)}(s \to t)$ indicate a targeted label-flipping attack.

F. Anomaly Reporting

Each client transmits a compact anomaly vector, which preserves privacy since neither raw data nor model gradients are shared:

$$\mathbf{v}_i^{(r)} = \left(\rho_i^{(r)}, \max_{s \neq t} \xi_i^{(r)}(s \to t)\right) \in [0,1]^2 \tag{19}$$

The server aggregates reports across selected clients $\mathcal{S}_r$ using a weighted average. The anomaly reports are weighted according to their anomaly confidence.

$$\mathbf{V}^{(r)} = \frac{\sum_{i \in \mathcal{S}_r} w_i^{(r)} \mathbf{v}_i^{(r)}}{\sum_{i \in \mathcal{S}_r} w_i^{(r)}}, \tag{20}$$

with aggregation weights:

$$w_i^{(r)} = \rho_i^{(r)} + \max_{s \neq t} \xi_i^{(r)}(s \to t) \tag{21}$$

Consequently, clients exhibiting stronger anomaly evidence contribute more significantly to the global anomaly assessment, while uncertain reports are naturally downweighted. Since malicious clients tend to minimize their reported anomaly confidence to evade detection. Their influence is naturally reduced by the weighting mechanism., their contributions are naturally suppressed by this weighting scheme, limiting the impact of spoofed reports. This compact, vectorized representation enables efficient and privacy-preserving anomaly detection [34].

G. Server-Side CBC

Local anomaly reports $\mathbf{v}_i^{(r)} \in [0,1]^2$ from clients are valuable, but they remain isolated alerts. To detect coordinated attacks, DualFed aggregates reports over a sliding window of W rounds using CBC. The CBC score accumulates temporally and cross-client aligned behavioral signals as follows:

$$\Psi_{s \to t}^{(r)} = \frac{\sum_{R=r-W+1}^{r} \sum_{i \in \mathcal{S}_R} w_i^{(R)} \, \xi_i^{(R)}(s \to t)}{\sum_{R=r-W+1}^{r} \sum_{i \in \mathcal{S}_R} w_i^{(R)}} \tag{22}$$

A coordinated attack is detected whenever $\max_{s,t} \Psi_{s \to t}^{(r)} \geq \tau_S$. By correlating temporal and semantic patterns in client reports, DualFed transforms distributed vigilance into a dual defense mechanism. Upon detection, DualFed triggers a silent self-recovery mechanism.

H. Non-Punitive Self-Recovery Mechanism

Once a coordinated poisoning attack is confirmed, DualFed neutralizes the embedded backdoor without excluding participating clients from the federation. The global model is decomposed into $\theta^{(r)} = \left(\theta_{\text{feat}}^{(r)}, \theta_{\text{cls}}^{(r)}\right)$, where θ_{feat} represents the feature extractor and θ_{cls} corresponds to the classifier head. Following the principle of representation stability, targeted label-flipping attacks mainly corrupt the decision boundary while preserving low-level feature representations. Accordingly, the feature extractor is preserved: $\theta_{\text{feat}}^{(r+1)} = \theta_{\text{feat}}^{(r)}$. To instantly eradicate the malicious decision boundary, the classifier head undergoes a complete hard reinitialization:

$$\theta_{\text{cls}}^{(r+1)} = \mathcal{I}_{\text{init}}(d_{\text{in}}, d_{\text{out}}), \tag{23}$$

where $\mathcal{I}_{\text{init}}$ denotes Kaiming initialization [9]. By fully resetting the corrupted layer while preserving the intact deep representations, the global model rapidly recovers its benign accuracy in the subsequent rounds. Let $\oplus$ denote parameter concatenation. The complete DualFed update rule is

$$\theta^{(r+1)} = \begin{cases} \theta_{\text{feat}}^{(r)} \oplus \mathcal{I}_{\text{init}}(d_{\text{in}}, d_{\text{out}}), & \max_{s,t} \Psi_{s \to t}^{(r)} \geq \tau_S, \\ \theta^{(r)} + \sum_{i \in \mathcal{S}_r} p_i \, \Delta\theta_i^{(r)}, & \text{otherwise} \end{cases} \tag{24}$$

IV. Experimental Evaluation

Our experiments use MNIST, CIFAR-10, and the Stroke Prediction dataset. The datasets are divided into training (80%) and testing (20%) sets. Data is partitioned under IID and non-IID settings. For MNIST, a quantity-skewed variant is also evaluated. The Stroke dataset's features are min–max scaled to $[0,1]$ and partitioned by age and stroke prevalence to induce heterogeneity. We simulate 100 clients with $f_{\text{mal}} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ malicious clients. At each round, 50% are selected. Malicious clients perform a coordinated label-flipping attack (7→4 on MNIST/CIFAR-10, 1→0 on Stroke). The label flipping attack is launched at round 4, with a threshold $\tau_S = 2.0$. We evaluate DualFed using three key metrics: (1) Malicious accuracy: represents the fraction of source-class inputs that the model incorrectly predicts. (2) Detection and mitigation rounds: the detection round corresponds to the first time the server identifies a coordinated attack, while the mitigation round is the immediate round (r+1). (3) Benign accuracy: measures the model's performance on clean test data. We additionally report the final benign accuracy after n rounds. All models are trained using SGD with a learning rate of 0.01, a local batch size of 10, and 5 local epochs. The model architectures are tailored to the data modality: For MNIST, we use a 2-layer CNN followed by two fully connected layers. For CIFAR-10, a deeper CNN with three fully connected layers is used. A 3-layer MLP (64→32→2) with ReLU activations and dropout ($p = 0.3$) is utilized in the case of the Stroke Prediction dataset.

A. Anomaly Detection Lifecycle

As illustrated in Fig. 2, each coordinated label-flipping attack follows a characteristic life cycle. The attack begins at round 4. Malicious accuracy reaches 100% while benign accuracy remains low (14–20%). Detection occurs at round 9, as DualFed requires temporally correlated anomaly reports to confirm a coordinated behavior. At round 10, the decision layer is reset. Immediately the malicious accuracy is reduced to 0%. Then, a rapid benign accuracy recovery from 20.36% to over 52% in a single round is achieved. In contrast, in the "No Defense" baseline, accuracy steadily drops below 16%.

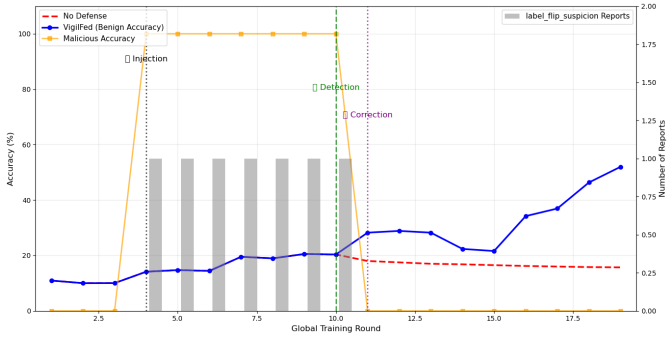

Fig. 2. Lifecycle of a Coordinated Label-Flipping Attack in a Non-IID FL System (CIFAR-10, CNN, 10% Malicious Clients).

B. Mitigation Robustness

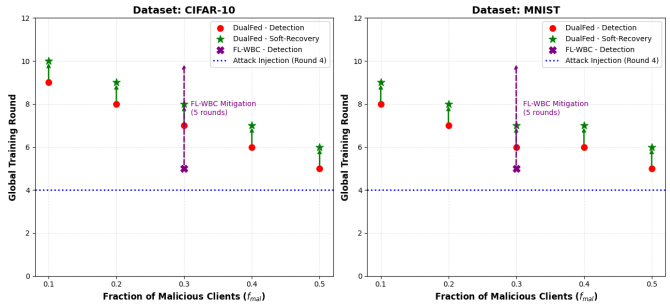

Fig. 3. Impact of the fraction of malicious clients on attack detection and mitigation timing.

The fraction of malicious clients (f_{mal}) directly influences the timing of DualFed's detection and mitigation. For each $f_{\text{mal}} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, we performed 5 independent runs. Then, we report the median detection round. As f_{mal} increases from 0.1 to 0.5, DualFed detects attacks earlier, shifting from round 8 to 5 on MNIST and from round 9 to 5 on CIFAR-10. This trend reflects the fact that stronger coordinated attacks are easier to detect. The slightly delayed detection observed on CIFAR-10 can be explained by its higher task complexity and noisier training dynamics. In contrast, FL-WBC detects attacks at round 5 but requires 5 additional rounds for full mitigation. DualFed applies a fast and immediate correction in the immediate next round as described in Fig. 3.

C. Comparative Evaluation

We evaluate DualFed against two recent defense mechanisms: FL-WBC[30], a client-based detection system, and CBDF[27], a clustering-based defense for coordinated attacks. In the comparison with FL-WBC, we focus on the evolution of benign accuracy under a coordinated label-flipping attack on MNIST and CIFAR-10. The attack begins at round 4. FL-WBC detects the anomaly at round 5 and mitigates it at round 12 through progressive stabilization. It reaches 81.0% on MNIST and 48.0% on CIFAR-10 as final benign accuracy as described in Fig. 4. In contrast, DualFed detects the coordinated attack at round 8 on both datasets. Mitigation begins immediately at round 9 via a targeted reset of the classification head. Despite detecting the attack at a later stage, DualFed achieves faster recovery. Moreover, DualFed maintains superior final accuracy, reaching 84.0% on MNIST and 52.0% on CIFAR-10 at round 15. These results demonstrate that DualFed enables more effective long-term model recovery. In table I a holistic comparison across security and privacy criteria is proposed. To ensure a fair evaluation, we adapt DualFed to the experimental settings used in [30] and[27] . For comparison with CBDF, the Stroke Prediction dataset was partitioned into 30 clients over 10 independent runs, with various fractions of malicious clients ($f_{\text{mal}} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$). The final accuracy is averaged across all runs. For FL-WBC, we followed the original setup with 100 clients and 5 runs. DualFed achieves coordinated attack detection while preserving client inclusivity and enabling rapid recovery. Unlike CBDF, which relies on client exclusion, DualFed detects attacks within 5 rounds and triggers silent self-recovery in the next round. It also restores the global model without removing any participant. In terms of final benign accuracy, DualFed achieves 95.7% on the Stroke Prediction dataset, outperforming CBDF's 91.0%, even though CBDF actively excludes malicious clients. In contrast, DualFed maintains high privacy by avoiding raw data inspection and relying solely on behavioral signals. The low detection delay and one-round recovery time demonstrate the efficiency of temporal correlation of DualFed

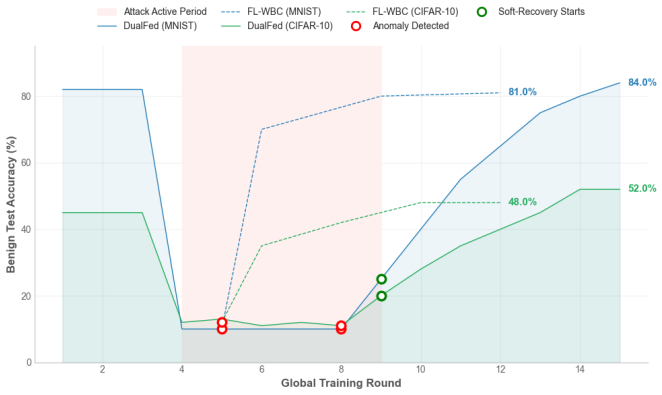

Fig. 4. Recovery Effectiveness Comparison between DualFed and FL-WBC under Non-IID Setting ($f_{\text{mal}} = 0.5$).

TABLE I
PERFORMANCE COMPARISON OF DUALFED WITH FL-WBC AND CBDF.

Criterion	**FL-WBC [30]**	**CBDF [27]**	**DualFed(Ours)**
Coordinated attack	No	Yes	Yes
Client Exclusion	No	Yes	No
Detection Delay	5 rounds	–	5 rounds
Recovery Time	High	High	1 round
Privacy Preservation	High	High	High
Final Accuracy : (Stroke prediction dataset)	–	91.0%	**95.7%**

and classifier reset mechanism. These results confirm that DualFed offers a superior trade-off between security, speed, inclusivity, and privacy.

V. Conclusion and Future Perspectives

In this work, we presented DualFed, a novel dual-side defense mechanism for detecting and mitigating coordinated label-flipping attacks in FL. DualFed leverages a CBC mechanism to expose synchronized malicious behavior. Local signals, such as PD and LFS, are monitored and aggregated over a sliding window. Upon detection, DualFed triggers a silent self-recovery mechanism. This approach ensures rapid model recovery, maintains system continuity, and upholds privacy. Extensive experiments on MNIST, CIFAR-10, and Stroke Prediction datasets demonstrate that DualFed achieves promising results compared to FL-WBC and CBDF. For future work, we envision DualFed as the core of a multi-layer immune system for FL. Inspired by [8], this system will integrate cross-client consistency checks, adaptive thresholds, and a memory module to store attack signatures. In addition, we will extend DualFed by integrating adaptive strategy learning [35] to mitigate coordinated adversarial behavior.

References

[1] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-Efficient Learning of Deep Networks from Decentralized Data. Proceedings of MLR, pp. 1273-1282, 2017.

[2] Muhammed, D., Ahvar, E., Ahvar, S., Trocan, M., Sinaie, M., and Ehsani, R. Federated learning for predicting irrigation requirements in multi-farm irrigation scheduling systems. In Intelligent Information and Database Systems, 2024.

[3] Hong, T. P., Chen, C. C., Chen, C. H., and Li, K. S. M. Privacy Preserving Based on SHA Encryption and Cluster Analysis in Federated Frequent Itemset Mining. In Asian Conference on Intelligent Information and Database Systems (pp. 83-94), 2025.

[4] Wan, Y., Qu, Y., Ni, W., Xiang, Y., Gao, L., Hossain, E.: Data and Model Poisoning Backdoor Attacks on Wireless Federated Learning, and the Defense Mechanisms: A Comprehensive Survey. IEEE Communications Surveys & Tutorials, pp. 1234–1256.IEEE (2024).

[5] Blanchard, P., Guerraoui, R., Stainer, J., Rouault, S.: Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent, Advances in Neural Information Processing Systems (NeurIPS), pp. 119–129,2017.

[6] Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: Proceedings of the 35th International Conference on Machine Learning, pp. 5650–5659 (2018).

[7] Nguyen, T.T., Pathirana, P.N., Ding, M., Seneviratne, A.: A Survey of Security Strategies in Federated Learning: Defending Models, Data, and Privacy. Lecture Notes in Computer Science (LNCS), 2022.

[8] Hamdani, H., Benmohamed, E., Ltifi, H.: AFA-DPD: Adaptive Federated Approach using Data Poisoning Detection. In: Proceedings of the 22nd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2025).

[9] Rossbroich, J., Gygax, J., Zenke, F.: Fluctuation-driven initialization for spiking neural network training. In: Neuromorphic Computing and Engineering, vol. 2, no. 4, pp. 44016,2022.

[10] Yadav, C., Bottou, L.: Cold Case: The Lost MNIST Digits. In: Advances in Neural Information Processing Systems (NeurIPS 2019), pp. 153–163. Curran Associates, Inc. (2019).

[11] Krizhevsky, A., Nair, V., Hinton, G.E.: Learning Multiple Layers of Features from Tiny Images. Technical Report, University of Toronto (2009).

[12] Fedesoriano, F.: Stroke Prediction Dataset. In: Kaggle Datasets (2020).

[13] Hu, K., Gong, S., Zhang, Q., Seng, C., Xia,M: An overview of implementing security and privacy in federated learning. *Artificial Intelligence Review*, 2024.

[14] Almutairi, S., Barnawi, A.: Federated Learning Vulnerabilities, Threats and Defenses: A Systematic Review and Future Directions. *Internet of Things*, vol. 24,2023.

[15] Elmahfoud, E., El Hajla, S., Maleh, Y., Mounir, S., Ouazzane, K.: Label flipping attacks in hierarchical federated learning for intrusion detection in IoT. *Information Security Journal: A Global Perspective*, 34(4), pp. 310–326,2025.

[16] Chen, M., Ning, Y., Li, H., Zhang, J., Wang, S.: Targeted Backdoor Attacks via Label Flipping in Federated Learning. *Neural Networks*, pp. 231–242,2023.

[17] Bhagoji, A.N., Chakraborty, S., Mittal, P., Calo, S.: Analyzing federated learning through an adversarial lens. In: Proceedings of the International Conference on Machine Learning,2019.

[18] Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., Shmatikov, V.: How to backdoor federated learning. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2020).

[19] Xie, C., Huang, K., Chen, P.Y., Li, B.: DBA: Distributed backdoor attacks against federated learning. In: Proceedings of the International Conference on Learning Representations (2020).

[20] Gong, X., Chen, Y., Huang, H., Liao, Y., Wang, S., Wang, Q.: Coordinated Backdoor Attacks against Federated Learning with Model-Dependent Triggers. In: IEEE Network, vol. 36, no. 1, pp. 84–90 (2022).

[21] Istiqomah, R.F., Sukarno, P., Wardana, A.A.: Coordinated attacks detection simulation with deep neural network algorithm and federated learning. In: 7th World Conference on Computing and Communication Technologies (WCCCT 2024), pp. 12–16. IEEE, April 2024.

[22] Barkatsa, S., Diamanti, M., Charatsaris, P., Voikos, S., Tsiropoulou, E.E., Papavassiliou, S.: Coordinated Jamming and Poisoning Attack Detection and Mitigation in Wireless Federated Learning Networks. In: IEEE Open Journal of the Communications Society (2025).

[23] Wardana, A.A., Sukarno, P.: Taxonomy and survey of collaborative intrusion detection system using federated learning. In: ACM Computing Surveys, vol. 57, no. 4, pp. 1–36 (2024).

[24] Zhang, G., Liu, H., Yang, B., Feng, S.: DWAMA: Dynamic weight-adjusted Mahalanobis defense algorithm for mitigating poisoning attacks in federated learning. In: Peer-to-Peer Networking and Applications, 2024.

[25] Deng, J., Liu, S., Li, C.: Detecting diverse poisoning attacks in federated learning based on joint similarity. In: 16th International Conference on Wireless Communications and Signal Processing, pp. 133–138, 2024.

[26] Jiang, Y., Zhang, W., Chen, Y.: Data quality detection mechanism against label flipping attacks in federated learning. In: IEEE Transactions on Information Forensics and Security, vol. 18, pp. 1625–1637 (2023).

[27] Anastasiadis, D., Refanidis, I.: Defending Federated Learning from Collaborative Poisoning Attacks: A Clique-Based Detection Framework. In: Electronics, vol. 14, no. 10, p. 2011 (2025).

[28] OlusolaOlarinde, M., Abiona, A.A., Ajinaja, M.O.: Adaptive and Privacy-Preserving Security for Federated Learning Using Biological Immune System Principles. In: Asian Journal of Computer Science and Technology, vol. 13, no. 2, pp. 67–72 (2024).

[29] Uprety, A., Rawat, D.B., Sadler, B.: Human immune system inspired security for federated learning-empowered Internet of Things. In: ACM Transactions on Internet of Things, vol. 3, no. 1, pp. 1–22 (2025).

[30] Sun, J., Li, A., DiValentin, L., Hassanzadeh, A., Chen, Y., Li, H.: FL-WBC: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. In: Advances in Neural Information Processing Systems 34 . Curran Associates, Inc., 2021.

[31] Zhai, R., Chen, X., Pei, L., Ma, Z.: A federated learning framework against data poisoning attacks on the basis of the genetic algorithm. In: Electronics, vol. 12, no. 3, p. 560 (2023).

[32] Lecué, G., Lerasle, M.: Robust machine learning by median-of-means: theory and practice. In: The Annals of Statistics, 48(2), pp. 906–931. Institute of Mathematical Statistics (2020).

[33] Yin, D., Chen, Y., Kannan, R., Bartlett, P.: Byzantine-robust distributed learning: Towards optimal statistical rates. In: Proceedings of the 35th International Conference on Machine Learning (ICML), pp. 5650–5659. PMLR (2018).

[34] Rousseeuw, P.J., Hubert, M.: Anomaly detection by robust statistics : Data Mining and Knowledge Discovery, 2018.

[35] Datar, Mandar, and Yann Dujardin. "Adaptive Learning for Moving Target defence: Enhancing Cybersecurity Strategies." *2025 11th International Conference on Control, Decision and Information Technologies (CoDIT)*. Vol. 1. IEEE, 2025.