IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2025, accepted 15 July 2025, date of publication 24 July 2025, date of current version 8 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3592207

APPLIED RESEARCH

FedChallenger: A Robust Challenge-Response and Aggregation Strategy to Defend Poisoning Attacks in Federated Learning

M. A. MOYEEN[1], (Graduate Student Member, IEEE), KULJEET KAUR[2,3,4], ANJALI AGARWAL[1], (Senior Member, IEEE), S. RICARDO MANZANO[5], MARZIA ZAMAN[6], AND NISHITH GOEL[5]

[1]Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada
[2]École de Technologie Supérieure (ÉTS), University of Quebec in Montreal, Montreal, QC H2X 3Y7, Canada
[3]Department of Electrical Engineering, Canadian University Dubai, Dubai, United Arab Emirates
[4]Centre for Research Impact and Outcome, Chitkara University, Rajpura, Punjab 140401, India
[5]Cistech Ltd., Ottawa, ON K2E 7K3, Canada
[6]Cistel Technology Inc., Ottawa, ON K2E 7V7, Canada

Corresponding author: M. A. Moyeen (ma.moyeen@concordia.ca)

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alliance through the collaboration of Concordia University, École de Technologie Supérieure (ÉTS), and Cistech Ltd., with project title "Secure Federated Learning at the Edge" under Grant ALLRP 568539-21.

ABSTRACT Growing data privacy concerns in smart applications have spurred the development of Federated Learning (FL), a novel approach enabling heterogeneous clients to jointly train a global model without exchanging private data. However, FL faces significant challenges in aggregating model updates from different client devices, as malicious participants can poison the data and model updates to corrupt the global model. To enhance the global model's accuracy, many state-of-the-art defence strategies in FL rely on aggregation-based security mechanisms. However, the global model can be more accurate if an attacker is excluded from the training. Therefore, this research proposes a dual-layer defence mechanism called *FedChallenger* to detect and prevent malicious client participation in the FL training process. The defence mechanism incorporates zero-trust challenge-response-based trusted exchange in the first layer, whereas, in the second layer, it uses a variant of the Trimmed-Mean aggregation strategy that uses pairwise cosine similarity along with Median Absolute Deviation (MAD) for aggregation to mitigate the malicious model parameters. Extensive evaluation using MNIST, FMNIST, EMNIST, and CIFAR-10 datasets demonstrates that the proposed *FedChallenger* outperforms state-of-the-art approaches, including Stake, Shap, Cluster, Trimmed-Mean, Krum, FedAvg, and DUEL, across both attack and non-attack scenarios. Under adversarial conditions with model and data poisoning attacks, *FedChallenger* achieves a 3-10% improvement in global model accuracy over the closest contender, along with 1.1-2.2 times faster convergence. Additionally, it attains a 2-3% higher F1-Score than the best-competing technique while maintaining robustness against varying attack intensities across different dataset complexities.

INDEX TERMS Federated learning, machine learning, poisoning attacks, robust aggregation, zero trust security.

I. INTRODUCTION

Intelligent and smart electronics applications are the result of the rapid development of the Internet of Things (IoT) and Artificial Intelligence (AI). Machine Learning (ML), the core of AI, has enabled the development of life-changing applications whose performance largely depends on the amount of training data. According to [1], IoT devices are projected to produce more than 600 zettabytes (ZB) of data by 2030 as the number of interconnected devices is expected to

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

reach 500 billion [2]. Although there is a promise of abundant data, increasing privacy concerns have begun to undermine the benefits of ML.

Federated Learning (FL) alleviates privacy concerns by keeping client data locally while enabling collaborative model training [3]. In FL, a central server coordinates multiple clients who train a shared model using their local data and exchange only model parameters. Despite FL's privacy benefits, model poisoning and data poisoning attacks are evolving rapidly [4], [5]. In model poisoning attacks, attackers manipulate model parameters to degrade performance, while in data poisoning attacks, they manipulate training labels to cause misclassification [6]. Both attacks can cause accuracy drops with potentially life-threatening consequences. In response, several defence mechanisms have been proposed, including FedAvg [7], Krum [8], Trimmed-Mean [9], and Fang et al.'s [10] defence strategy combining ERR and LFR approaches, hereafter referred to as DUEL. FedAvg naively averages all client updates without discrimination, making it vulnerable to basic poisoning attempts. Krum selects updates closest to their neighbours via Euclidean distance minimization, but fails against coordinated attacks where multiple adversaries craft seemingly legitimate updates. Trimmed-Mean mitigates outliers by discarding extreme values based on median deviation, but its fixed trimming ratio cannot adapt to dynamic poisoning strategies. DUEL introduces parameter-wise loss impact analysis but employs reactive strategies that only mitigate attacks after compromising the training process.

A. MOTIVATION

Most state-of-the-art approaches defend against poisoning attacks using either robust aggregation [8], [9], [10] or verification-based rejection [11], [12]. However, these approaches still lack accuracy and cannot fully defend against poisoning attacks, as they rarely offer any mechanism to prevent attacks from propagating into the aggregation computation. Recent hybrid defence techniques, such as Stake [13], Shapley-based methods [14], and Cluster-based approaches [15], hereafter referred to as Stake, Shap, and Cluster respectively, face significant challenges including high computational costs and reduced accuracy with Non-IID data distributions. Stake utilizes blockchain technology for majority voting, incorporating client reward and penalty calculations, which enhances security but introduces significant processing delays. Shap relies on SHAP value computations, creating substantial computational demands, especially for complex models. Cluster analyzes gradients from client updates to identify source and target classes before applying HDBSCAN clustering [16] to detect malicious updates, which proves particularly sensitive to variations in data distribution. Since state-of-the-art techniques do not guarantee a poisoning-free environment, a robust defence mechanism capable of detecting, preventing, and efficiently recovering from such attacks remains critical.

B. CONTRIBUTIONS

This paper proposes an extension of our previously designed *FedChallenger* approach [17] that can detect and prevent malicious participants from participating in FL training using challenge-response mechanisms. The extended approach introduces a revised zero-trust challenge-response architecture [18] that actively authenticates all participating devices before and during training sessions. The framework incorporates an improved robust aggregation algorithm utilizing Median Absolute Deviation (MAD)-based [19] trimming to enhance resilience against poisoning attacks. Furthermore, the performance of the extended version is evaluated across multiple benchmark datasets [20], including comparisons with the most recent techniques [13], [14], [15].

The major contributions of this paper are as follows.

- Propose a zero-trust challenge-response-based defence mechanism named *FedChallenger* to detect and prevent poisoning attacks on consumer electronic devices.
- Present trust-based attack detection algorithm that relies on challenge-response information to compute trust.
- Introduce a robust aggregation mechanism that applies cosine similarity-based consensus boosting to benign weights and dynamically prune malicious updates via MAD for adversarial updates.
- Evaluate the performance of the proposed approach on MNIST, FMNIST, EMNIST, and CIFAR-10 datasets [20] using different evaluation metrics such as convergence time and accuracy.

C. ORGANIZATION

The rest of the paper is organized as follows: The next section presents relevant state-of-the-art literature. Section III illustrates the design and architecture of the *FedChallenger*. The experimental setup is presented in Section IV. After that, the results and discussions are discussed. Finally, in section VI, the manuscript is concluded with future research directions.

II. RELATED WORK

This section presents relevant literature on the proposed *FedChallenger* approach that defends against poisoning attacks. A significant amount of research has already been conducted to defend against model and data poisoning attacks. Verification and aggregation-based strategies [11], [12] show notable performance among them. Verification strategies rely on detecting any alterations in model updates using ML or cryptography techniques. However, the robust aggregation mechanisms [8], [9], [10] focus on the aggregation function to detect and discard outlier samples. This section provides a brief overview of the state-of-the-art pertinent literature focusing on both robust aggregation and verification-based strategies.

A. AGGREGATION-BASED STRATEGIES

The ultimate goal of poisoning attackers is to degrade the model's accuracy by modifying the model parameters or data

labels. Inaccuracies in model predictions can cause drastic life-threatening scenarios or induce huge financial losses. Therefore, poisoned updates should be removed before evaluating the trained model. In response to the problem, Bulyan [21], Trimmed-Mean [9], Krum [8], FABA [22], and Fang et al.'s schemes [10] introduced novel robust aggregation strategies to detect and remove malicious updates. Most of these strategies, including Krum, Bulyan, and Trimmed-Mean, rely on the Euclidean distance to remove malicious samples. Krum considered model updates malicious if the computed Euclidean distance exceeded a threshold value. Bulyan computed the Euclidean distance between model updates and considered the model updates closest to their median value. It chooses them for aggregation and computes their mean to consider them one of the model parameters. However, Euclidean distance is often skewed by the alteration of a single model parameter [10]. Therefore, Krum and Bulyan share the same attack space and cannot fully defend themselves from poisoning attacks.

Trimmed-Mean [9] with a given trimmed rate, discards the smallest and the largest distance values after sorting them in ascending order. The mean of the remaining values is calculated to determine the model parameters. However, the distance calculation is mainly based on the Euclidean distance and can be vulnerable to similar attack types, such as Krum and Bulyan. Another fast aggregation strategy, FABA [22], introduced iterative pruning of model updates that are distant from average model updates. However, it is still dependent on distance calculations, and distance often gets largely changed due to changes in a single parameter. Therefore, Fang et al.'s scheme [10] can remove outliers by considering model updates that cause a loss in the model evaluation and negatively impact the model accuracy. Liu and colleagues [23] introduced an alternative method that leverages the Pearson correlation coefficient. By calculating the dissimilarity between malicious and benign model parameters using this coefficient, their technique enables the detection and subsequent removal of harmful updates.

Defence techniques proposed in [24], [25], and [26] remove training samples that induce a higher error rate in model evaluation. Another strategy, TRIM [27], minimizes the loss function and infers the training subset using the given model parameter. If the data sample did not yield the inferred subset, it discarded that training sample, assuming it was a poisoned update. However, these approaches cannot fully detect and prevent the impact of malicious model updates on the final aggregation because of their limited understanding of the entire dataset.

B. VERIFICATION-BASED STRATEGIES

Verification-based strategies [11], [12] have emerged to identify poisoned updates in communication rounds. One such approach, BAFFLE [12], uses a feedback-based learning mechanism to decide on poisoned updates with other validating clients collaboratively. However, validating clients can be malicious, and attackers may use other datasets. Another verification-based strategy uses a digital signature [28] to ensure a trusted exchange of information. This strategy detects any alterations in the model updates by leveraging digital signatures.

Generative Adversarial Networks (GAN) [29] are extremely popular and have emerged as game changers in the Artificial Intelligence (AI) market. These strategies can be used to generate an auditing dataset that can train a classifier to predict malicious samples [30]. However, GAN solutions are proven to behave differently across different data distributions. Therefore, they show different characteristics in Independent and Identically Distributed (IID) and Non-IID datasets [31], [32]. In addition to generating auditing datasets, a reference model is often used to predict model parameters [33]. These predicted parameters can be used to replace malicious parameters. However, constructing a reference model is vigorous and may not always comply with the desired accuracy. Malicious updates are often distinguished from benign samples using Support Vector Machine (SVM) classifiers [34]. To distinguish malicious updates from benign updates, FLARE [35] utilizes a penultimate layer representation vector. The computed results offer a trust score for local model updates, which can determine the approval or rejection of model updates. However, their trust computation often requires significant initial information [36].

C. RECENT TECHNIQUES

Among the notable contributions of recent advancements in poisoning attacks are Stake [13], Shap [14], and Cluster [15] techniques. Stake uses blockchain [13] for aggregation, where clients update their local updates to a blockchain, and the aggregation process happens in the blockchain. The aggregated updates are validated by designated voters, who vote for acceptance or rejection of the updates. If the majority of them accept the update, then voters and proposers are rewarded. However, the unaccepted updates, voters, and the proposer are slashed. Though the technique is robust, it might have significant computational complexity due to the nature of the blockchain. On the other hand, the Shap technique uses SHAP values, which leave a visible mark of poisoning attacks on the feature space. However, they require a reference dataset for SHAP computation, which is against FL's privacy guarantee. The cluster technique uses source and target neurons as discriminative features for label-flipping attack detection, and using that information, the HDB-SCAN [16] cluster can differentiate malicious and benign samples. However, clustering techniques are limited to common clustering problems, and the feature space analysis requires some knowledge about the dataset.

While methods like FLGuardian [37] have pioneered layer-wise defence mechanisms using cosine similarity or Euclidean distance analysis and weighted trust scoring to detect anomalous updates by comparing pairwise

similarities across neural network layers, they face challenges in handling adaptive poisoning attacks that strategically manipulate gradients to evade distance-based detection. The reliance on static clustering algorithms may fail against dynamic attacks that gradually shift malicious updates to mimic benign patterns, especially in Non-IID settings where natural layer-wise variations exist. Additionally, such methods struggle with high computational overhead when scaling to complex models, as pairwise comparisons across all layers and clients become prohibitively expensive. In contrast, AIDFL [38] introduces a novel information-theoretic framework that leverages conditional entropy and mutual information metrics, which are inherently independent of data distributions, to detect poisoning attacks by examining the structural relationships between data and model layers. Unlike traditional methods that employ static clustering or aggregation rules, AIDFL implements a multi-level defence protocol combining K-means clustering [39] with dynamic anomaly detection based on information flow patterns across network layers. While AIDFL's information-theoretic approach effectively handles Non-IID data, its reliance on mutual entropy calculations incurs higher computational overhead. Additionally, AIDFL lacks explicit client authentication to verify the legitimacy of updates.

Another defence mechanism named MSGuard [40] combines sign statistics, cosine similarity, and spectral anomaly scores in a Mean Shift clustering model to detect Byzantine attacks without prior knowledge of attacker counts. However, its reliance on gradient magnitude filtering may inadvertently discard benign updates in Non-IID settings. Additionally, the computational overhead of multi-feature clustering could hinder scalability in large-scale FL systems. In contrast, TDF-PAD [41] uses IQR to classify models as poisoned, benign, or ambiguous, then applies Z-score analysis to ambiguous cases. Its adaptive thresholds enhance Non-IID robustness, although computational costs may slow convergence, and dynamic attacks may evade detection. Another defence mechanism called PurifyFL [42] combines homomorphic encryption with poisoning attack detection via cosine direction analysis of updates. While its single-server design enhances practicality by supporting additive and multiplicative ciphertext operations, the approach may inadvertently filter benign updates due to directional thresholds and impose computational burdens on resource-constrained devices. However, FLAD [43] advances the state-of-the-art by introducing neural Feature Extraction Models (FEM) trained on server data to enable adaptive gradient feature analysis through DBSCAN clustering while simultaneously addressing privacy via CKKS homomorphic encryption [44]; however, its effectiveness depends on the representativeness of the server's clean dataset and may struggle against sophisticated adaptive poisoning attacks that mimic legitimate gradient patterns or exploit the reduced feature space.

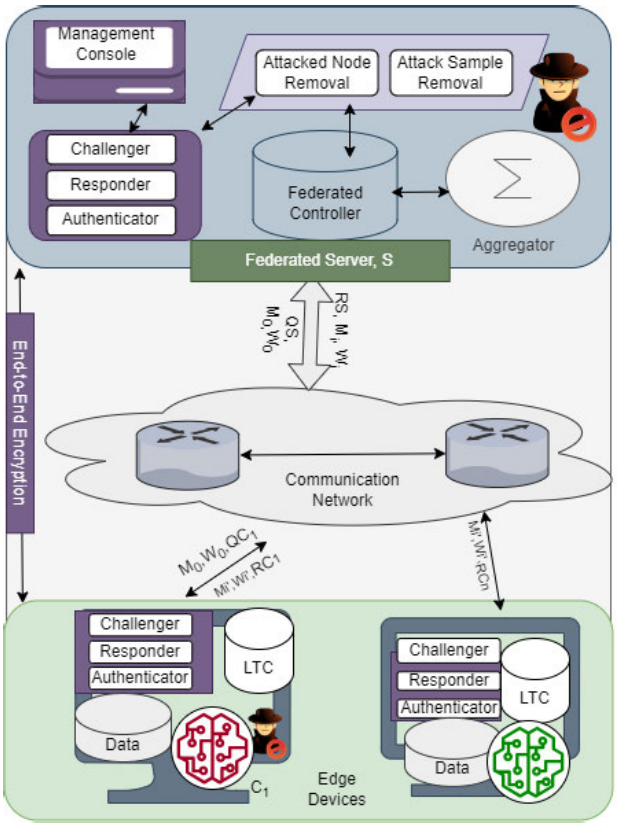


FIGURE 1. The architecture of fedchallenger.

All recent techniques struggle with computational complexity and may often be confined to a particular dataset type.

D. WHY FEDCHALLENGER?

Existing verification-based rejection or robust aggregation strategies can defend against a limited set of attacks. Moreover, they cannot entirely block the propagation of attacks into the aggregated global model. Furthermore, recent state-of-the-art strategies mostly prune benign samples and incur significant computational overhead. Thus, *FedChallenger* is proposed to offer a lightweight multi-layer defence. It is expected to filter out the most malicious transactions during the challenge-response phase and remove the remaining malicious updates using a robust aggregation strategy. Above all, FedChallenger incorporates MAD-based adaptive pruning, which mainly filters out non-benign records.

III. DESIGN

This section presents the design and architecture of the proposed FedChallenger technique for mitigating poisoning attacks.

FedChallenger follows a modular zero-trust [18] architecture where the core components are the Challenger, Responder, Authenticator, and Aggregator. Figure 1 presents the architecture of the proposed FedChallenger strategy.

Algorithm 1 Malicious Device Detection

Input:
Decrypted Challenge, $C'_{i,S}$
Shared secret ($K_{i,S}$), Challenge Matrix (CM), Expected Results ($ER_{i,S}$), Similarity Threshold (ST), Trust Factor (TF) Trust Score ($T_{i,S}$) & Trust Threshold (TTH)
Output:
Malicious Status (M_i)

Selected Challenge, $CS \in Rand(CM, Total(CM))$
for each $c \in CS$ **do**
  Encrypted Challenge, $EC_{i,S} = K_{i,S}(c)$
  Response, $R = challengedEntity(EC_{i,S})$
  Expected Response, $C'_{i,S} = ER_{i,S}(c)$
  Response Similarity, $C_{sim} = similarity(C', R)$
  if $C_{sim} > ST$ **then**
    $T_{i,S} = T_{i,S} \times TF$
  else
    $T_{i,S} = T_{i,S}/TF$
  end if
end for
if $T_{i,S} > TTH$ **then**
  $M_i = False$
else
  $M_i = True$
end if
Return M_i

The architecture organizes the Federated server, S, Communication Network, CN, and Edge Devices, ED, in a top-down layout. In the considered setup, each ED and S contains dedicated challenge-response modules that consist of Challenger, Responder, and Authenticator submodules. The Challenger submodule asks questions to the target device, and the Responder submodule answers the question asked by the Challenger submodule of another device. The Authenticator submodule makes decisions and authenticates devices based on their challenge-response exchange. The management console at S defines the criteria for the questions to be asked. The federated controller, FC, coordinates model training, removes attacker nodes, eradicates attack samples, aggregates model updates, and implements management decisions. Attacker node and attack sample removal are coordinated with the FC, challenge-response module and their respective submodules. Each ED has a Local Training Controller, LTC and a neural engine. They use the initial model parameters and obtain training information from the FC. Using this initial information, their LTC coordinates a model training using their respective local data. The trained model parameters are then returned to the S for aggregation. The FC aggregates model parameters after receiving from multiple participants using the aggregator submodule. If the FC finds any model updates or the devices are malicious, they remove them immediately after detection to prevent their impact on the training. The aggregator submodule can also filter out the outliers originating from malicious participants.

Algorithm 2 Training With Defence at Client, i

Input:
Batches, B
Model, M'
Weights, W'
Output:
Weight (W'_B)

Training Request, $TRR_i = RequestToParticipate(S)$
Run Parallel $ChallengeResponseModule()$
Shared Secret, $K_{i,S} = establishSharedSecret()$
if TRR_i **then**
  if $isMAlicious(S, K_{i,S})$ **then**
    $ReduceTrustScore(S)$
  end if
  for $b \in B$ **do**
    $W'[b] = W'[b-1] \cup MiniBatchSGD(M'_i, W'_i)$
  end for
end if
Return W'_B

At the beginning of the training process, the management console instructs the FC to broadcast the details of the model training and asks it to accumulate participants from the EDs. The interested participants $I = (i_1, i_2, \ldots, i_n) \in ED$ responds to the FC's request and establishes a shared secret key, $K_{i,S}$ for their subsequent communication. Subsequently, each interested participant, i and S, ask themselves a set of management-defined questions, $Q_{i,S}$ and $Q_{S,i}$, respectively, to accumulate their respective challenge matrix $CM_{i,S}$ and $CM_{S,i}$ for future challenges. The questions encompass a range of system attributes, such as CPU speed, memory capacity, fan speed, data folder size, and the hash values of data points at designated random indices. All participants have their peer's challenge matrix, $CM_{i,S}$ and $CM_{S,i}$ consisting of unaltered challenge-response data. In later stages, this information is used to detect a change in the participant's behaviour, which can eventually lead to the early detection of malicious participants. With the questions and answers, the FC begins the training rounds.

In the first round, it challenges the participants with random questions $Q_{S,i'}$ from its question pool, and each participant answers those questions $R_{i',S}$ and asks back questions $Q_{i',S}$ to the S. FC matches the answer with its accumulated challenge matrix CM_S. In response to a mismatch, FC removes the participants from its accumulated participant list and blocks further interaction in coordination with the attacker node removal submodule. For the rest of the participants, the FC responds with the answer $R_{S,i'}$ for each of their questions and sends the initial model M_0, weights W_0. Upon receiving this information, each of the I, at first verifies the server response $R_{S,i'}$ with their stored $CM_{i',S}$ and if they find any mismatch, they can abstain from participating in the training process. Otherwise, they use the initial model M_0, and weights W_0 to begin the training. The LTC at each I uses those initial parameters and their local data to train the model

Algorithm 3 Incorporated Defence at Federated Server, S

Input:
Initial Weights (W_0), Communication Rounds (R), Loss threshold (LTH), & Initial Model (M_0)
Output:
Global Model (M)

Broadcast training request, $TRR_{S,*}$
Run Parallel, $ED = getInterestedParticipants()$
for each $r \in R$ **do**
  Random Picked Clients, $I = randPickClientset(ED)$
  for $i \in I$ **do**
    if $r = 0$ **then**
      Shared Secret, $K_{i,S} = establishSharedSecret()$
      $W_{i,r-1} = W_0$
      $M_{i,r-1} = M_0$
    else
      if $isNotMalicious(i, K_{i,S})$ **then**
        $W_i, r \cup Train(W_{i,r-1}, M_{i,r-1}, K_{i,S})$
      end if
    end if
  end for
  Cosine Similarity Between Weights, $X_{S,I,r} = cosSim(W_{I,r})$
  $W_{I',r} = PickmWeights(W_{I,r}, X_{S,i,r})$
  $W^a_{I,r} = FedAVG(W_{I',r})$
  $M_{I,r} = Load(M_{I,r-1}, W^a_{I,r})$
  $L_{I,r} = ComputeLoss(W^a_{I,r}, M_{I,r})$
  if $L_{I,r} > LTH$ **then**
    $Discard(W^a_{I,r})$
  else
    $M = M \cup M_{I,r}$
  end if
end for
Return M

with a pre-established algorithm such as Stochastic Gradient Descent (SGD) [45].

A. MALICIOUS DEVICE DETECTION

Algorithm 1 presents the malicious device detection strategy for all client and server devices. For malicious client detection, S challenges I with $CS \in CM$ to receive its response, R. Each challenge $c \in CS$ is encrypted with a shared secret between client and server $K_{i,S}$ before transmission to ensure the confidentiality and integrity of the challenge. After that, the S computes the similarity between R and Expected Response, $C'_{i,S}$ for the respective challenge, $c \in CS$, where $ER_{i,s}$ module accumulates the responses. The higher similarity compared to a management-defined threshold, ST leads to the multiplicative increase of trust by a trust factor TF; otherwise the trust is penalized by the same factor. After computing the overall trust for each CS, the trust threshold determines the device's malicious status.

B. CLIENT-SIDE DEFENCE

The client-side defence and training processes are depicted in Algorithm 2.

The algorithm runs challenge-response modules to facilitate the challenge-response-based defence through a question-answering analogy. At first, the client, i, expresses its interest in participating in the training to the S in the form of a training request TRR_i. The acceptance of the request succeeds with the establishment of shared secrets, $K_{i,S}$ and initial model-related information. During the training process, malicious activity detection using Algorithm 1 results in a reduction in the trust score of S and the training halts; otherwise, the mini-batch SGD determines the model weights W' for each batch, $b \in B$. Finally, the weights are returned to the S for further processing.

C. DEFENCE AT FEDERATED SERVER

Algorithm 3 presents the dual-layer defence and federated training process at S.

Here, a background broadcast of training requests, TRR_s results in an accumulation of interested devices, EDs. The accumulation process and broadcasts continued to be run to facilitate the required number of participants. At each round of communication, a random subset of client devices, I, is selected, and Algorithm 1 is run in parallel to detect and remove malicious participants. In the initial round of communication, M_0, W_0, and shared secret key $K_{i,S}$ are established at each training participant; where the management decides on the M_0, W_0 and $K_{i,S}$ is established through Diffie—Hellman exchange [46]. Only non-malicious clients defined by the Algorithm 1 are allowed to train the model, and their weights are accumulated in the $W_{i,r}$. At the end of each communication round, the cosine similarity among client weights, X_S is computed using the Eq. (1).

$$X_S[j,j'] = \frac{\boldsymbol{W}_j \cdot \boldsymbol{W}_{j'}}{\|\boldsymbol{W}_j\| \|\boldsymbol{W}_{j'}\|}, \quad \forall j,j' \in \{1,\ldots,n\},\ j \neq j'. \tag{1}$$

Subsequently, Eq. (2) determines the weights $W_{I',r}$ for federated averaging (FedAvg) to obtain W^a.

$$\boldsymbol{W}_{I',r} = \begin{cases} \{\boldsymbol{W}_j + \lambda \cdot \mathrm{sim}_j \cdot (\boldsymbol{W}_j - \boldsymbol{W}_{\mathrm{med}})\}_{j=1}^{n}, & \max(\mathrm{MAD}_j) \leq \gamma \\ \{\boldsymbol{W}_{(j)}\}_{j=1}^{m}, & \text{otherwise} \end{cases} \tag{2}$$

where
$$\mathrm{MAD}_j = \operatorname*{med}_{k'} \left| X_S[j,k'] - \mathrm{med}(X_S) \right|,$$
$$\gamma = \mathrm{med}(\mathrm{MAD}_{k'}) + \mathrm{Var}(\mathrm{MAD}_{k'}),$$
$$m = \left\lfloor n \cdot \frac{\mathrm{med}(\mathrm{MAD}_{k'})}{\mathrm{med}(\mathrm{MAD}_{k'}) + \mathrm{Var}(\mathrm{MAD}_{k'})} \right\rfloor,$$
$$\mathrm{sim}_j = \mathrm{med}_{k'}(X_S[j,k']),$$
$$\boldsymbol{W}_{\mathrm{med}} = \text{element-wise median of } \{\boldsymbol{W}_j\}_{j=1}^{n}$$

Under normal conditions (when $\max(\mathrm{MAD}_j) \leq \gamma$ for all clients $j \in n$), the weights are consensus boosted through the operation $W_j + \lambda \cdot \mathrm{sim}_j \cdot (W_j - W_{\mathrm{med}})$, where $\gamma = \mathrm{med}(\mathrm{MAD}_k) + \mathrm{Var}(\mathrm{MAD}_k)$ and $\mathrm{sim}_j = \mathrm{med}_k(X_S[j,k])$ represents client j's median cosine similarity. In adversarial conditions, the system selects $m = \left\lfloor n \cdot \frac{\mathrm{med}(\mathrm{MAD}_{k'})}{\mathrm{med}(\mathrm{MAD}_{k'}) + \mathrm{Var}(\mathrm{MAD}_{k'})} \right\rfloor$ weights by sorting all weights in ascending order of their MAD_j values (where k' indexes the weight vectors).

After that, the averaged weights are loaded into the current model state $M_{I,r-1}$ to compute the loss $L_{I,r}$. This significant

loss results in discarding the model updates. The final model M is achieved after the model reaches the desired accuracy.

D. THEORETICAL ANALYSIS

This subsection presents the theoretical proof of robustness and convergence of the proposed algorithms.

Theorem 1 (Robustness Against Poisoning Attacks): The FedChallenger design is robust against poisoning attacks due to its challenge-response mechanism and malicious device detection algorithm. Specifically, the probability of a malicious participant maintaining a trust score T_i above the threshold T_{thresh} after d challenges is bounded by:

$$P(T_i \geq T_{\text{thresh}}) \leq \left(\frac{1}{\alpha}\right)^{d-\log_\alpha(T_{\text{init}}/T_{\text{thresh}})},$$

where $\alpha > 1$ is the trust factor, T_{init} is the initial trust score, and T_{thresh} is the trust threshold.

proof: The following mechanisms ensure the robustness of FedChallenger:

1. **Challenge-Response Mechanism:** Each participant i is challenged with a set of questions $Q = \{q_1, q_2, \ldots, q_n\}$. The expected response r_j for each challenge q_j is computed as $r_j = f(q_j)$, where $f(\cdot)$ is a deterministic function known only to legitimate devices. A malicious participant providing incorrect responses $r'_j \neq r_j$ is detected with high probability, as the probability of guessing all responses correctly is:

$$P_{\text{malicious}} = \prod_{j=1}^{n} P(r'_j = r_j) \leq \left(\frac{1}{|\mathcal{R}|}\right)^n,$$

where $|\mathcal{R}|$ is the size of the response space.

2. **Trust Score Dynamics:** The trust score T_i of participant i is updated as:

$$T_i \leftarrow \begin{cases} T_i \cdot \alpha, & \text{if } r'_j = r_j \quad \text{(correct response)}, \\ T_i/\alpha, & \text{if } r'_j \neq r_j \quad \text{(incorrect response)}, \end{cases}$$

where $\alpha > 1$ is the trust factor. Let T_{init} be the initial trust score. After l incorrect responses, the trust score becomes:

$$T_i = T_{\text{init}} \cdot \alpha^{-l}.$$

The participant is marked as malicious if $T_i < T_{\text{thresh}}$. Solving for l, we get:

$$l > \log_\alpha(T_{\text{init}}/T_{\text{thresh}}).$$

Thus, the number of incorrect responses required to mark the participant as malicious is:

$$l_{\text{required}} = \lceil \log_\alpha(T_{\text{init}}/T_{\text{thresh}}) \rceil.$$

The probability that a malicious participant provides fewer than l_{required} incorrect responses out of d challenges is bounded by:

$$P(T_i \geq T_{\text{thresh}}) \leq \left(\frac{1}{\alpha}\right)^{d-\log_\alpha(T_{\text{init}}/T_{\text{thresh}})}.$$

3. **Federated Aggregation with Cosine Similarity:** The server computes the cosine similarity matrix X_S for the weights $W = \{w_1, w_2, \ldots, w_n\}$ submitted by participants:

$$X_S[i,j] = \cos(w_i, w_j) = \frac{w_i \cdot w_j}{\|w_i\|\|w_j\|}.$$

The server filters out outliers by selecting m benign weights using $m = \left\lfloor n \cdot \frac{\text{med}(\text{MAD}_{k'})}{\text{med}(\text{MAD}_{k'}) + \text{Var}(\text{MAD}_{k'})} \right\rfloor$. Where, MAD_j is computed using

$$\text{MAD}_j = \underset{k'}{\text{med}} \left| X_S[j,k'] - \text{med}(X_S) \right|.$$

This ensures that only weights from non-malicious participants are included in the federated averaging process. Thus, the FedChallenger design is robust against poisoning attacks. □

Theorem 2 (Convergence of FedChallenger): The FedChallenger design converges to a global model that minimizes the loss function, assuming that most participants are non-malicious and the learning rate is appropriately chosen.

proof: The following steps ensure the convergence of FedChallenger:

1. **Local Training Process:** Each non-malicious participant i performs local training using mini-batch SGD. The updated rule for the weights w_i in next round denoted by $r+1$ is:

$$w_{i,r+1} = w_{i,r} - \eta_r \nabla \mathcal{L}_i(w_{i,r}),$$

where η_r is the learning rate and $\mathcal{L}_i(w)$ is the local loss function. Under standard assumptions (e.g., $\mathcal{L}_i(w)$ is Lipschitz smooth and convex) [47], mini-batch SGD converges to a local minimum of $\mathcal{L}_i(w)$.

2. **Federated Averaging:** The server aggregates the weights $W = \{w_1, w_2, \ldots, w_n\}$ from non-malicious participants using federated averaging:

$$w^a = \frac{1}{|W|} \sum_{i=1}^{|W|} w_i.$$

By the convexity of $\mathcal{L}_i(w)$, the federated average w^a satisfies:

$$\mathcal{L}(w^a) \leq \frac{1}{|W|} \sum_{i=1}^{|W|} \mathcal{L}_i(w_i),$$

where $\mathcal{L}(w)$ is the global loss function.

3. **Global Model Update:** The global model is updated iteratively as:

$$w_{r+1} = w_r - \eta_r \nabla \mathcal{L}(w_r).$$

Under the assumption that the majority of participants are non-malicious, the global model converges to a minimum of $\mathcal{L}(w)$.

4. **Loss-Based Filtering:** At each communication round, the server computes the loss $\mathcal{L}(w^a)$ and discards updates if $\mathcal{L}(w^a) > \mathcal{L}_{\text{thresh}}$, ensuring that only meaningful updates are applied to the global model.

Thus, the FedChallenger design ensures convergence to a global model that minimizes the loss function. □

TABLE 1. Dataset sample distribution.

Dataset	Total Samples	Training	Test	Validation
MNIST	60,000	45,000	10,000	5,000
FMNIST	60,000	45,000	10,000	5,000
CIFAR-10	60,000	45,000	10,000	5,000
EMNIST	382705	3,00,000	50,832	31,873

Lemma 1 (Trust Score Dynamics): The trust score T_i of a participant i decreases exponentially with the number of incorrect responses. Specifically, after l incorrect responses, the trust score becomes:

$$T_i = T_{\text{init}} \cdot \alpha^{-l},$$

where T_{init} is the initial trust score and $\alpha > 1$ is the trust factor.

Proof: The trust score T_i is updated as:

$$T_i \leftarrow T_i/\alpha \quad \text{(for each incorrect response)}.$$

After l incorrect responses, the trust score becomes:

$$T_i = T_{\text{init}} \cdot \alpha^{-l}.$$

Thus, the trust score decreases exponentially with the number of incorrect responses. □

E. SECURITY ANALYSIS OF CHALLENGE-RESPONSE

The FedChallenger framework employs a mutual secret key ($K_{i,S}$)-based challenge-response mechanism to defend against poisoning attacks in FL. Thus, the challenge-response mechanism's security depends fundamentally on the confidentiality of $K_{i,S}$ and the unpredictability of generated challenges. The encryption of challenges via $EC_{i,S} = K_{i,S}(c)$ provides information-theoretic security when $K_{i,S}$ has a key length of $\lambda \geq 128$-bit. The probability of compromising $K_{i,S}$ with a brute-force attack is given by the Eq. (3).

$$\Pr[\text{Compromise } K_{i,S}] \leq \epsilon(\lambda) \approx 2^{-\lambda}, \tag{3}$$

where $\in (\lambda)$ represents the negligible success probability.

Furthermore, the unpredictability of challenge-response is related to the random generation of $CS \in CM$. Where CM combines multiple entropy sources, including the device's static, dynamic, and ephemeral traits. The static traits include, but are not limited to, device fingerprints, historical patterns, and so on. The dynamic traits, on the other hand, include recent activity logs, temporal usage patterns, and so on. Ephemeral information consists of session-specific nonces. The probability that the attacker guesses the i-th challenge, c_i correctly, given their knowledge (A) about the participant, is given by Eq. (4).

$$\Pr[\text{Successful guess}] = \prod_{i=1}^{n} \Pr[c_i \in CS | \mathcal{A}]. \tag{4}$$

Morever, Without knowledge of $K_{i,S}$ the probability is $\Pr[c_i \in CS] \approx \frac{1}{|CM|}$. Even if the attacker has partial CM knowledge, the probability $\Pr[c_i \in CS]$ is negligible. Furthermore, the trust score mechanism provides adaptive protection. Thus, the challenges and responses are secure enough for most cases.

IV. EXPERIMENTAL SETUP

This section presents the experimental setup for evaluating the proposed FedChallenger technique's resilience against model and data poisoning attacks. All simulations were conducted on a MacBook M1 Pro system with an 8-core CPU, 14-core GPU, 16GB RAM, and 512GB SSD. The chosen model for training was a Residual Neural Network (ResNet-18), whose architecture begins with an initial 7×7 convolutional layer applying 64 filters with stride two and 'same' padding, followed by batch normalization, ReLU activation, and 3×3 max-pooling. The network then processes data through four sequential residual blocks: the first two blocks employ two 3×3 convolutional layers with 64 and 128 filters, respectively, each preceded by batch normalization and ReLU, while the third and fourth blocks expand to 256 and 512 filters with identical kernel configurations, incorporating skip connections to mitigate vanishing gradients. The resulting feature maps are globally averaged and processed through a fully connected classification head comprising a single linear layer with softmax activation for multi-class prediction. Training employs mini-batch stochastic gradient descent (SGD) with a learning rate of 0.001 and batch size of 20, executing five local epochs per client across 100 clients with Non-IID data partitioning.

This section presents the results and discussion of the performance evaluation of the FedChallenger compared to FedAvg, Krum, Trimmed-Mean, DUEL, Stake, Shap, and Cluster approach. The overall evaluation results have been subdivided into three scenarios: no-attack scenario, model poisoning, and data poisoning attacks.

The Non-IID evaluation framework has been created using the MNIST, FMNIST, EMNIST, and CIFAR-10 datasets as referenced in [20] for comprehensive dataset-wise evaluation. These four datasets were specifically chosen because they represent different levels of complexity and are widely recognized benchmarks in FL research. MNIST provides a fundamental baseline with its simple grayscale digit images, while FMNIST increases the challenge with fashion item classification at the exact resolution. EMNIST extends this further by adding handwritten letters to the digit classification task. Finally, CIFAR-10 introduces colour images and more complex object recognition scenarios. Together, these datasets enable rigorous testing across various data distributions and difficulty levels, ranging from basic shape recognition to more sophisticated visual classification tasks. Their standardized formats and everyday usage in the FL literature ensure fair comparisons with existing methods while covering the essential spectrum of Non-IID data challenges. Table 1 presents the dataset sample distribution where MNIST, FMNIST, and CIFAR-10 have 60,000 samples for each of them and 45000 samples were used for the training set, 10,000 samples were applied as test samples, and 5,000 samples were used as validation samples. For the EMNIST dataset, 300,000 samples have been used for the

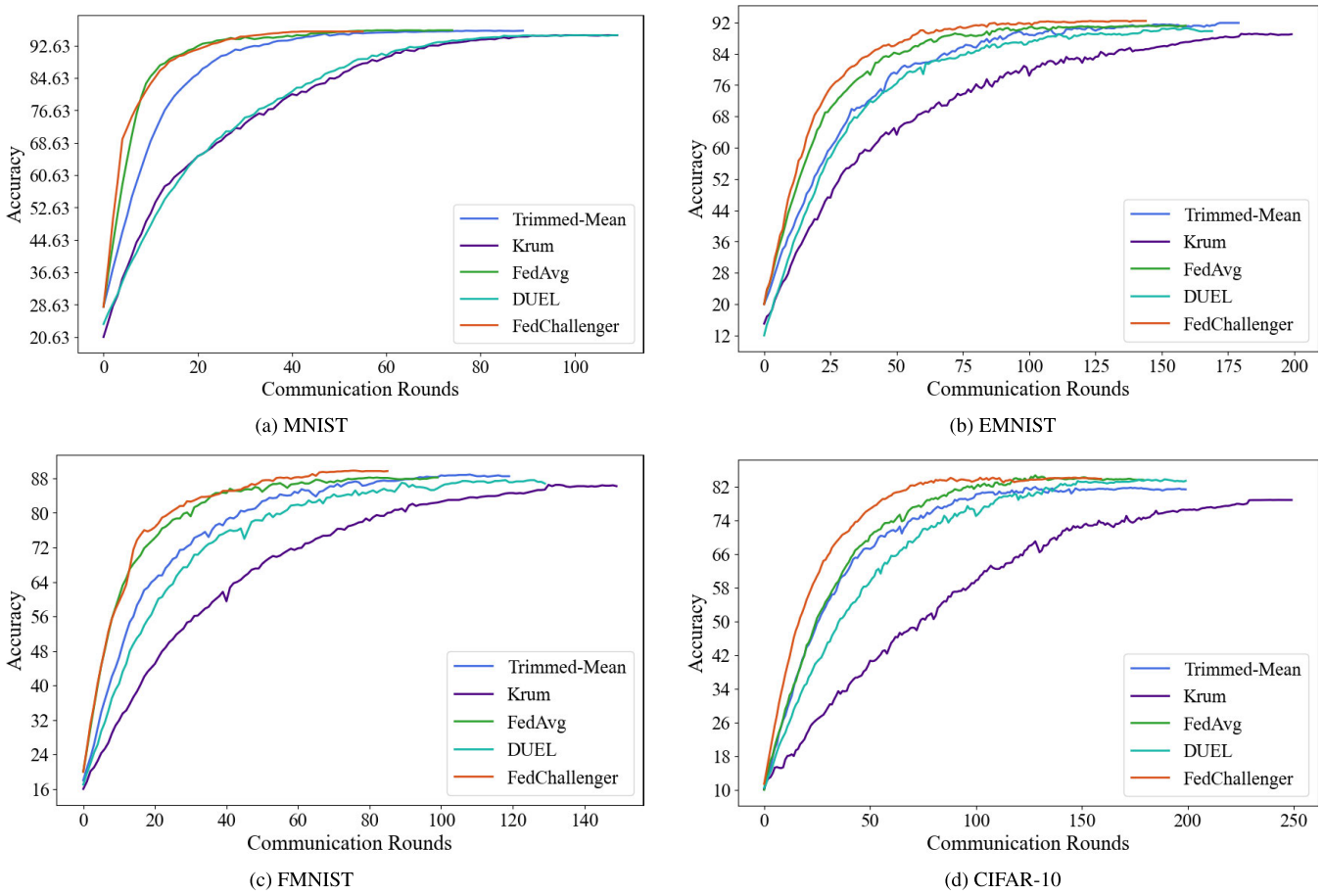


FIGURE 2. No attack scenario: Evaluation of accuracy.

training set, 50,832 samples have been used for the test set, and 31,873 samples have been used for validation. During the Non-IID data creation process, every client has been assigned two classes of data as per McMahan et al.'s. [48] approach. Furthermore, the error rate has been considered the early stopping criterion for the simulation. To simulate a model poisoning attack, we considered the most popular Gaussian attack [49] where the j^{th} model parameter has been replaced with a number drawn from a Gaussian distribution. On the other hand, the data poisoning attack has been simulated considering the well-researched label-flipping attack [15]. To simulate this attack, the original data label was randomly replaced with an alternative label from within the dataset.

FedChallenger's evaluation has been conducted under different scenarios, including i) no-attack, ii) Model Poisoning attack, and iii) Data Poisoning attack. The Gaussian attack is considered in the model poisoning attack, and *FedChallenger*'s percentage accuracy has been computed under different percentages of compromised devices. In the no-attack scenario, the accuracy has been measured to highlight the robustness of the proposed approach. Moreover, the average convergence time measurement highlights the efficacy of the *FedChallenger* in a no-attack scenario. In a data poisoning attack, model accuracy was measured by different percentages of data label-flipping. The presence of different percentages of compromised devices has determined the label-flipping percentage. To show the superiority of the proposed FedChallenger, the aforementioned evaluation scenarios consider comparison with DUEL approach [10], Trimmed-Mean [9], Krum [8], FedAvg [7], Stake [13], Shap [14], and Cluster [27] techniques.

V. RESULTS AND DISCUSSIONS

A. NO-ATTACK SCENARIO

Figure 2 compares the test accuracy of FedChallenger against baseline approaches, including FedAvg, Krum, Trimmed-Mean, and DUEL approach across MNIST, FMNIST, EMNIST, and CIFAR-10 datasets under no-attack scenarios. FedChallenger's innovative cosine similarity-based aggregation mechanism enables it to achieve competitive accuracy with fewer communication rounds while providing superior stability, particularly in Non-IID settings.

On MNIST evaluation, FedChallenger performs comparable to FedAvg and Trimmed-Mean, surpassing 96% accuracy within the initial training rounds. In contrast, Krum's method and DUEL approach requires additional communication rounds to attain their peak accuracy of 95%.

This performance gap stems from Krum's limitation of relying exclusively on a single client update per round, severely restricting gradient diversity. Meanwhile, DUEL approach suffers computational inefficiency due to its dual rejection mechanisms combining Loss Function Rejection (LFR) with Error Rate Rejection (ERR). FedChallenger's comparable performance emerges from its consensus boosting of updates that maintain strong directional alignment with the global model through cosine similarity, ensuring stable convergence while preserving all beneficial updates.

For FMNIST evaluations, FedChallenger establishes a clear accuracy advantage of 1.5%, 4.1%, 1.3%, and 3.4% over Trimmed-Mean, Krum, FedAvg, and DUEL approach, respectively, while demonstrating significantly steeper initial learning curves. Krum, Trimmed-Mean, and DUEL approach exhibits slower adaptation to FMNIST's complex feature space, whereas FedChallenger's similarity-driven weighting mechanism successfully balances contributions across diverse clients. This capability proves particularly valuable in reducing noise from Non-IID data distributions. The comparative approaches of Krum and DUEL exhibit substantially flatter learning curves due to their respective limitations - Krum's excessive rigidity in update selection and DUEL tendency toward unnecessary update discarding.

The EMNIST evaluation reveals FedChallenger maintaining performance parity with FedAvg and Trimmed-Mean while achieving significant accuracy improvements of 3.9% over Krum and 2.9% over DUEL approach. FedChallenger's exceptional stability under Non-IID conditions originates from its consensus boosted weights and rigorous enforcement of gradient alignment through cosine similarity metrics. This approach effectively prevents destabilization from skewed local updates, a vulnerability particularly apparent in FedAvg and Trimmed-Mean due to their lack of explicit geometric consistency verification mechanisms.

In the CIFAR-10 dataset, FedChallenger continues to match the accuracy levels of FedAvg and DUEL approach while substantially outperforming Krum by 6.4% and Trimmed-Mean by 3.1%. The early convergence in Figure 2 demonstrates FedChallenger's particular efficiency in high-dimensional spaces, where Krum's similarity-based selection criteria fail to maintain adequate gradient diversity. Trimmed-Mean struggles with CIFAR-10 properties and its blind threshold-based rejection, while FedChallenger's lightweight cosine-based weighting maintains scalability and robustness.

A critical observation from all datasets shows FedChallenger exhibiting markedly lower accuracy fluctuation under Non-IID conditions compared to baseline methods, as demonstrated in Figure 2. This stability advantage derives from FedChallenger's gradient alignment enforcement strategy, which intelligently weights updates according to their directional consistency. This approach mitigates client drift without ignoring Trimmed-Mean's aggressive pruning or Krum's problematic over-reliance on single updates. In contrast, FedAvg's simple uniform averaging inevitably accumulates variance from divergent clients, while DUEL rejection-based methodology unnecessarily discards potentially valuable updates in benign operational environments. Furthermore, FedChallenger incorporates MAD-based adaptive pruning of gradients, and that ensures zero filtered records for the no-attack scenario.

B. MODEL POISONING ATTACK SCENARIO

In the model poisoning attack scenario, the target device's model parameters are replaced with values drawn from a Gaussian distribution. Figure 3 illustrates the accuracy comparison of FedChallenger, Krum, Trimmed-Mean, FedAvg, and DUEL approach across successive iterations under this attack scenario. The evaluation considers the replacement of 40% of clients' model parameters with Gaussian-distributed values, with tests conducted on MNIST, FMNIST, EMNIST, and CIFAR-10 datasets to demonstrate FedChallenger's robustness.

The results reveal significant accuracy inconsistencies in Krum and FedAvg, establishing them as the worst-performing methods. FedChallenger demonstrates substantial accuracy improvements across all datasets, achieving 2.9, 3.2, 3.68, and 6.7 times higher accuracy than FedAvg on MNIST, EMNIST, FMNIST, and CIFAR-10 respectively. Compared to DUEL approach, FedChallenger maintains accuracy gains of 1.41, 1.54, 1.66, and 2.38 times across the same datasets. Furthermore, FedChallenger achieves absolute accuracy improvements of 2.65%, 14.24%, 11.15%, and 24.85% over Trimmed-Mean on MNIST, FMNIST, EMNIST, and CIFAR-10, respectively. Krum's Euclidean distance-based aggressive outlier removal proves particularly vulnerable, yielding 1.9, 2.6, 1.4, and 3.6 times lower accuracy than FedChallenger on the respective datasets. The evaluation conclusively demonstrates FedChallenger's consistent superiority across all datasets. Its dynamic m update selection, based on MAD-based estimation, enables adaptive pruning of updates while ensuring cosine similarity prioritizes update direction over absolute values. This approach makes FedChallenger resilient to Gaussian noise and skewed gradients caused by Non-IID data.

The evaluation framework assesses FedChallenger's robustness across increasing percentages of compromised devices. Figure 4 illustrates the accuracy trends from 0% to 30% poisoning rates, where adversarial clients inject Gaussian-distributed noise into model parameters. Consistent with theoretical expectations, all baseline methods demonstrate progressive accuracy degradation as poisoned updates increase.

FedChallenger maintains superior performance across all datasets, achieving accuracy improvements of 2.2, 2.6, 2.5, and 4.9 times relative to FedAvg in MNIST, EMNIST, FMNIST, and CIFAR-10 evaluations. Compared to DUEL approach, FedChallenger exhibits relative accuracy gains of 23.6%, 32.8%, 35.4%, and 69.7% across the same datasets.

While Trimmed-Mean emerges as the strongest competitor, FedChallenger maintains consistent advantages of 5.2%,

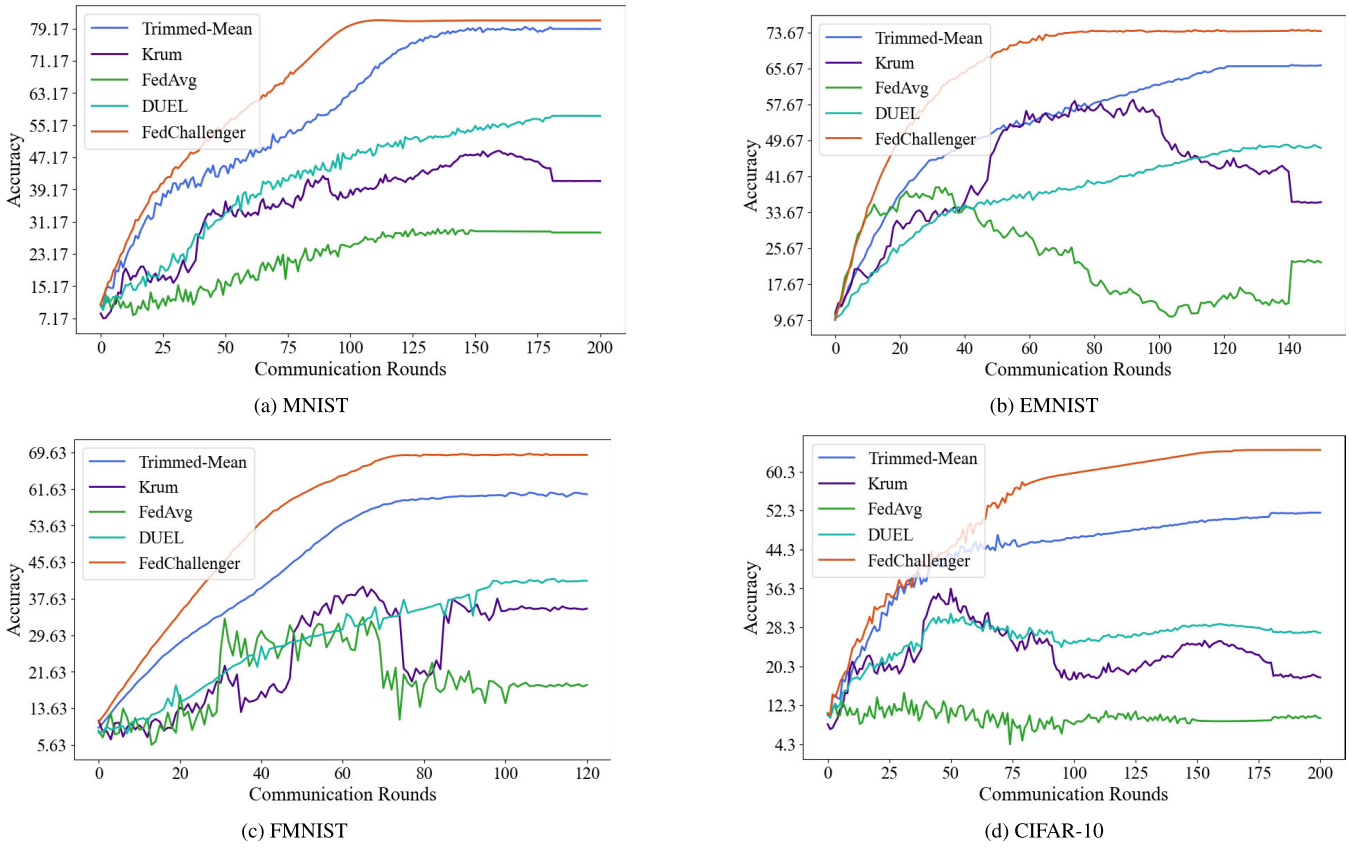


FIGURE 3. Model poisoning attack scenario: evaluation of accuracy under 40% compromised devices.

8.3%, 7.8%, and 17.2% in MNIST, FMNIST, EMNIST, and CIFAR-10 evaluations respectively. This performance advantage stems from FedChallenger's advanced aggregation strategy, which builds upon Trimmed-Mean by incorporating cosine similarity for enhanced robustness. Specifically, it dynamically prunes updates via MAD computation, adjusting the m value to enhance resilience further.

Krum's fundamental limitation of selecting only a single client update per aggregation round proves particularly detrimental, resulting in severe accuracy deficits of 40.1%, 55.2%, 51.2%, and 83.8% compared to FedChallenger across FMNIST, EMNIST, and CIFAR-10 evaluations. These comprehensive experimental results conclusively demonstrate FedChallenger's superior accuracy and resilience across all tested scenarios and datasets.

C. DATA POISONING ATTACK SCENARIO

In this evaluation, the target label of the training set is replaced with a randomly selected label from the available list, excluding the current label. Figure 5 illustrates the data poisoning attack scenarios, primarily focusing on random label-flipping. The evaluation examines cases where 0% to 30% of the clients' labels are flipped. The results demonstrate that FedChallenger consistently outperforms other methods. As the percentage of label-flipping increases, the accuracy of Trimmed-Mean declines sharply. Specifically, under a 30% compromised device scenario, Trimmed-Mean experiences an accuracy drop of 4.6% on MNIST, 7.6% on EMNIST, 8.2% on FMNIST, and 7.2% on CIFAR-10.

Similarly, while Krum maintains consistency in model poisoning and no-attack scenarios, its performance becomes unstable as label-flipping increases. FedChallenger achieves significantly higher accuracy than Krum, with improvements of up to 47.2% on MNIST, 61.5% on CIFAR-10, 47.8% on FMNIST, and 49.7% on EMNIST when 30% of device labels are flipped. Furthermore, compared to FedChallenger, FedAvg and DUEL approach exhibits substantial accuracy degradation under 30% label-flipping. FedAvg suffers drops of 93.5% on CIFAR-10, 56.4% on MNIST, 88.4% on FMNIST, and 80.4% on EMNIST, while DUEL method shows declines of 5.6% on CIFAR-10, 7.1% on MNIST, 7.4% on FMNIST, and 7.8% on EMNIST.

Despite 30% of data labels being flipped, FedChallenger maintains superior performance, demonstrating significantly higher accuracy than competing methods. These findings confirm that FedChallenger exhibits robustness against model and data poisoning attacks, attributable to its challenge-response-based mechanism and cosine similarity-based weight selection. The challenge-response mechanism dynamically verifies the credibility of participant updates, and cosine similarity with an adaptive m value pruning utilising MAD eliminates inconsistent updates.

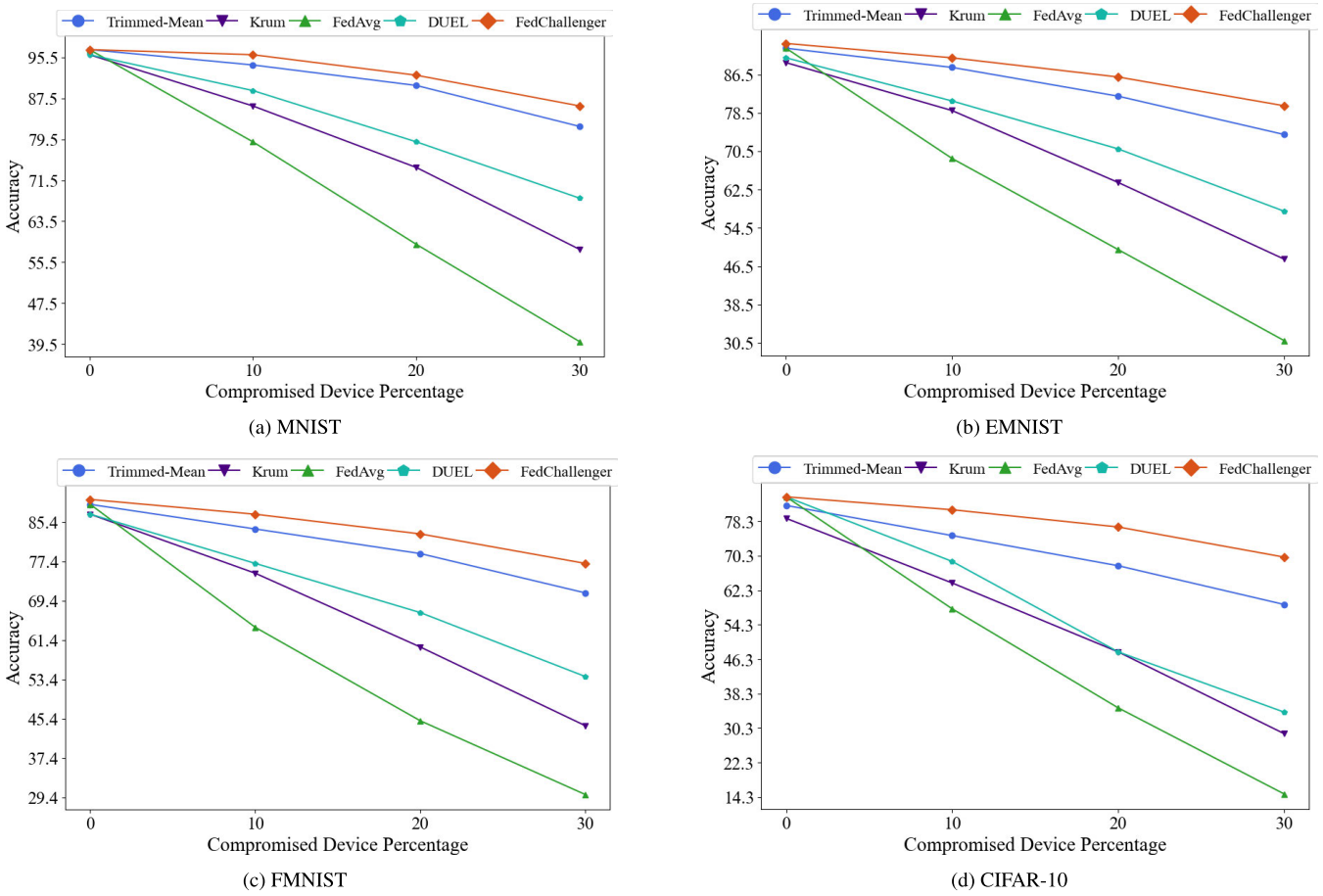


FIGURE 4. Model poisoning attack scenario: Evaluation of accuracy under different compromised device percentages.

Convergence Guarantee: In addition to improving accuracy, FedChallenger guarantees the fastest convergence among baseline approaches across varying levels of model and data poisoning attacks. Figure 6 compares the average convergence times of FedChallenger, Trimmed-Mean, Krum, FedAvg, and DUEL approach on MNIST, EMNIST, FMNIST, and CIFAR-10 datasets with 20% of client data poisoned.

On the MNIST dataset, FedChallenger achieves a 19.5% reduction in convergence time compared to Trimmed-Mean and outperforms DUEL method by 32.2%. The performance advantage is more pronounced when compared to FedAvg and Krum, showing improvements in convergence time of 57.1% and 56.2%, respectively. This substantial performance gap originates from FedChallenger's efficient cosine similarity-based aggregation mechanism, which effectively filters malicious updates while preserving valid gradient contributions.

The FMNIST evaluation yields similar results, with FedChallenger converging 24.2% faster than Trimmed-Mean and 17.8% faster than DUEL approach. The baseline methods FedAvg and Krum demonstrate significantly slower convergence, requiring 55.9% and 49.4% more time, respectively, to reach comparable accuracy levels. EMNIST experiments further validate FedChallenger's robustness, showing convergence time improvements of 53.7% over FedAvg and 52.5% over Krum. Compared to more sophisticated approaches, FedChallenger maintains a 19.2% advantage over Trimmed-Mean and an 18.6% improvement over DUEL method.

The consistent performance across datasets highlights the effectiveness of FedChallenger's aggregation strategy in handling varying data complexities. The most challenging CIFAR-10 evaluation reveals FedChallenger's strongest performance advantages, particularly against vulnerable baseline methods. FedChallenger converges 60.0% faster than FedAvg and 52.0% faster than Krum. When compared to more robust approaches, it maintains significant leads of 31.2% over Trimmed-Mean and 27.4% over DUEL This demonstrates FedChallenger's scalability to complex, high-dimensional data spaces while preserving its convergence advantages.

The comprehensive evaluation across MNIST, FMNIST, EMNIST, and CIFAR-10 demonstrates that FedChallenger achieves consistent convergence time improvements ranging from 17.8% to 60.0% compared to existing approaches.

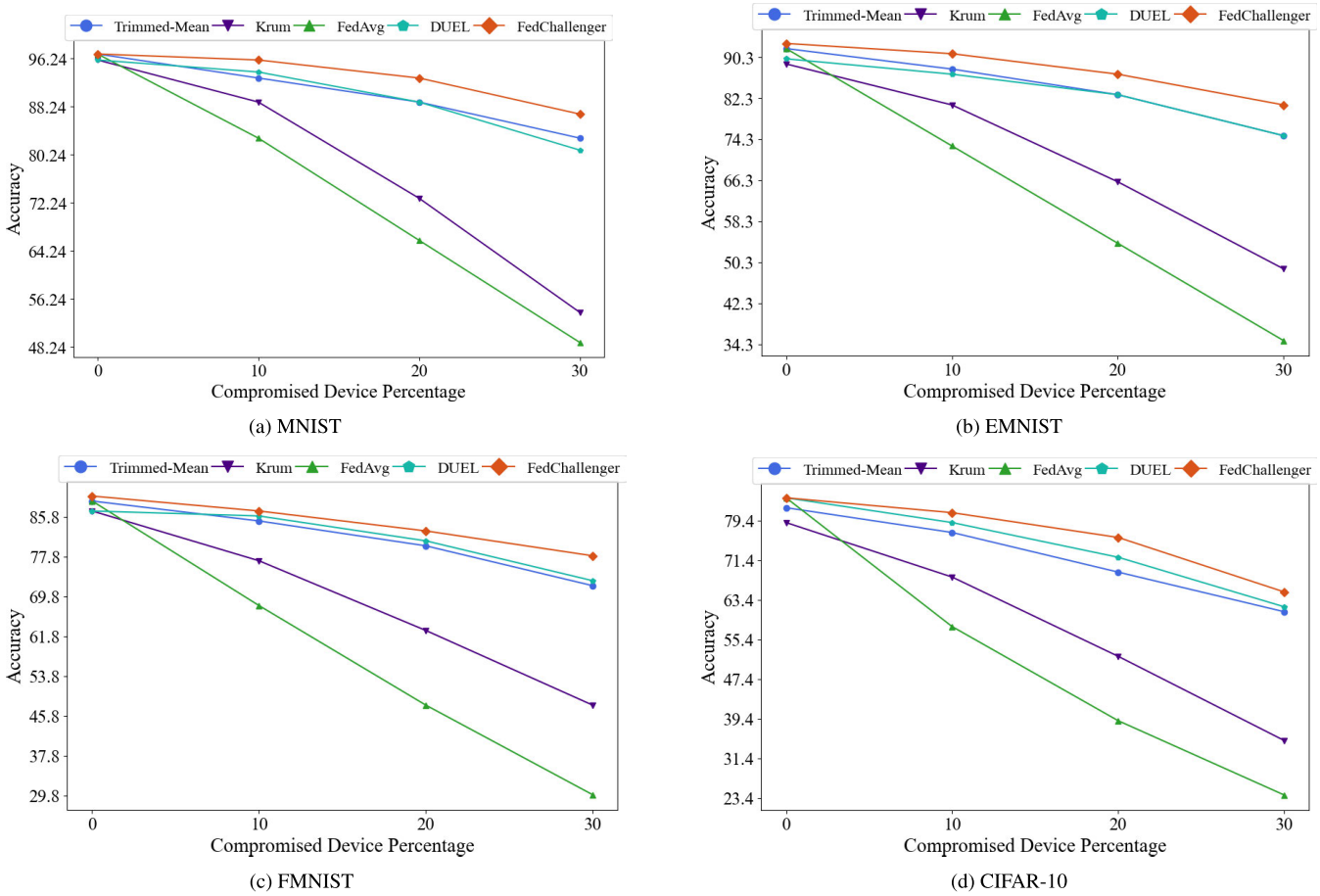


FIGURE 5. Data poisoning attack scenario: Evaluation of accuracy under different compromised device percentages.

These results demonstrate that FedChallenger's background challenge-response and light-weight cosine similarity-based aggregation utilising MAD computation achieves both faster convergence and enhanced robustness against poisoning attacks across diverse datasets and complexity levels.

D. PERFORMANCE COMPARISON WITH RECENT TECHNIQUES

To assess how FedChallenger performs against recent defence solutions—including Stake [13], Shap [14], and Cluster [15] in mitigating poisoning attacks, an extended evaluation was conducted for both data and model poisoning scenarios.

The Stake aggregation mechanism employs blockchain-based majority voting along with client reward and penalty computations. In this evaluation, Stake was simulated using an Ethereum smart contract implementation [50] with $\epsilon = 0.5$ in Web3.py [13]. On the other hand, Shap utilizes SHAP values to detect poisoning attacks; here, 10% of test samples were used as the background dataset for SHAP value computation. Meanwhile, the Cluster technique extracts source and target classes from gradients derived from clients' local updates, subsequently applying HDBSCAN [16] to cluster and identify malicious updates.

For the model poisoning attack, the parameters of a randomly selected device were replaced with a Gaussian distribution. In contrast, the label-flipping attack randomly replaced training data labels with other available class labels. The experimental setup remained consistent with prior evaluations to ensure a fair comparison. The results include convergence time and accuracy comparisons under data and model poisoning attacks.

1) CONVERGENCE TIME GUARANTEE

Figure 7 presents the average convergence time for Stake, Shap, Cluster, and FedChallenger Techniques under the No attack scenario to highlight their efficacy in normal operational mode. The experimental result suggests that datasets and size differences impact model convergence time.

In MNIST evaluation, Shap takes almost double the time to converge than FedChallenger because of SHAP value computation complexity, and Stake consumes nearly 63% more time due to the computation in Blockchain. Cluster, on the other hand, uses HDBSCAN and takes nearly 38% more time than FedChallenger. FedChallenger employs a lightweight challenge-response mechanism in the background while utilizing cosine similarity-based consensus boosting prior to weight averaging. This design yields

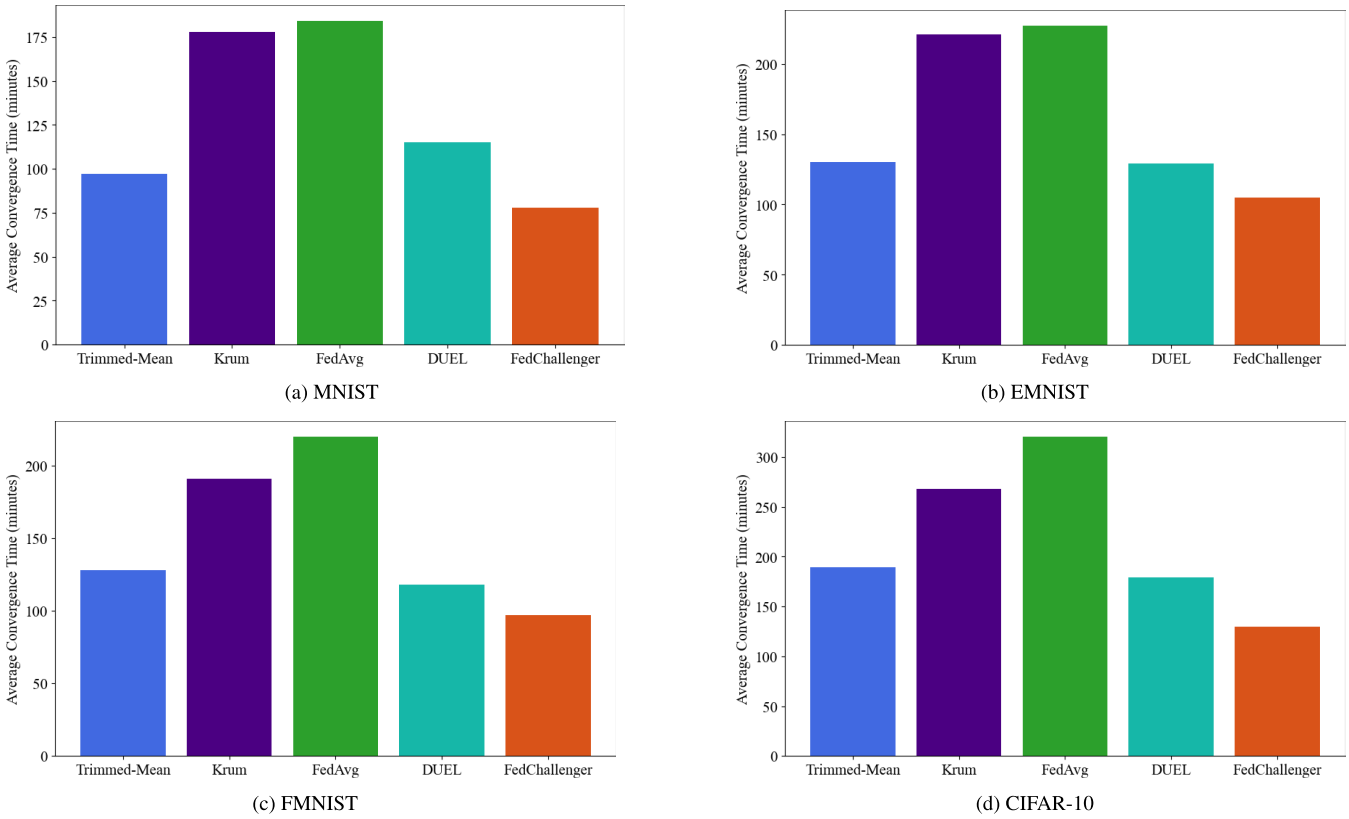


FIGURE 6. 20% data poisoning attack scenario: Evaluation of average convergence time.

significantly lower computational complexity than Stake, Shap, and cluster-based techniques.

Further evaluation with EMNIST, FMNIST, and CIFAR-10 datasets reveals that the FedChallenger is faster among its peers. In EMNIST, FedChallenger is 2.2, 2.4, and 2.5 times faster than the Stake, Shap, and Cluster techniques. Meanwhile, in the FMNIST evaluation, FedChallenger remains 1.8, 1.6, and 1.4 times faster than the Stake, Shap, and Cluster techniques, respectively.

In CIFAR-10 evaluation, FedChallenger becomes 46%, 42%, and 50% fastest than the Stake, Shap, and Cluster techniques, respectively. The results proved that the FedChallenger is a lightweight alternative to the recent state-of-the-art solutions.

2) MODEL POISONING ATTACK

This study comprehensively evaluates model poisoning resilience by systematically comparing the FedChallenger and Stake aggregation methods using the MNIST, EMNIST, FMNIST, and CIFAR-10 datasets. Other defence mechanisms were excluded from this evaluation as they primarily address data poisoning scenarios rather than model poisoning attacks. The experimental framework assumes that 40% of the client model parameters are compromised through adversarial manipulation.

The results, as presented in Figure 8, demonstrate distinct performance characteristics between the two defence mechanisms. FedChallenger exhibits superior resilience in MNIST and CIFAR-10 environments, achieving measurable accuracy improvements of 0.5% and 1.1%, respectively, compared to the Stake method. Conversely, Stake shows marginally stronger performance on EMNIST and FMNIST datasets, with accuracy advantages of 2.4% and 9.5%, respectively. These differential outcomes suggest a nuanced relationship between dataset characteristics and defence mechanism efficacy. FedChallenger's consistent performance across multiple evaluation scenarios, particularly its strong showing in more complex CIFAR-10 environments, indicates robust generalization capabilities.

3) DATA POISONING ATTACK

In the data poisoning attack scenario, class labels were randomly replaced with different labels, ensuring that each original label and its replacement were distinct. The evaluation considers MNIST, EMNIST, FMNIST, and CIFAR-10 datasets, and the compromised device percentage has been varied between 10% to 40%. Figure 10 presents the result of data poisoning attacks under different compromised device percentages in various datasets.

In MNIST evaluation, FedChallenger shows negligible performance improvement over the Shap technique. However, Stake and Cluster show nearly 3% and 6% drop in accuracy, respectively, in 40% compromised device scenarios. In contrast, the EMNIST evaluation in Cluster technique shows nearly a 6% drop in accuracy compared to FedChallenger under 40% compromised device percentage.

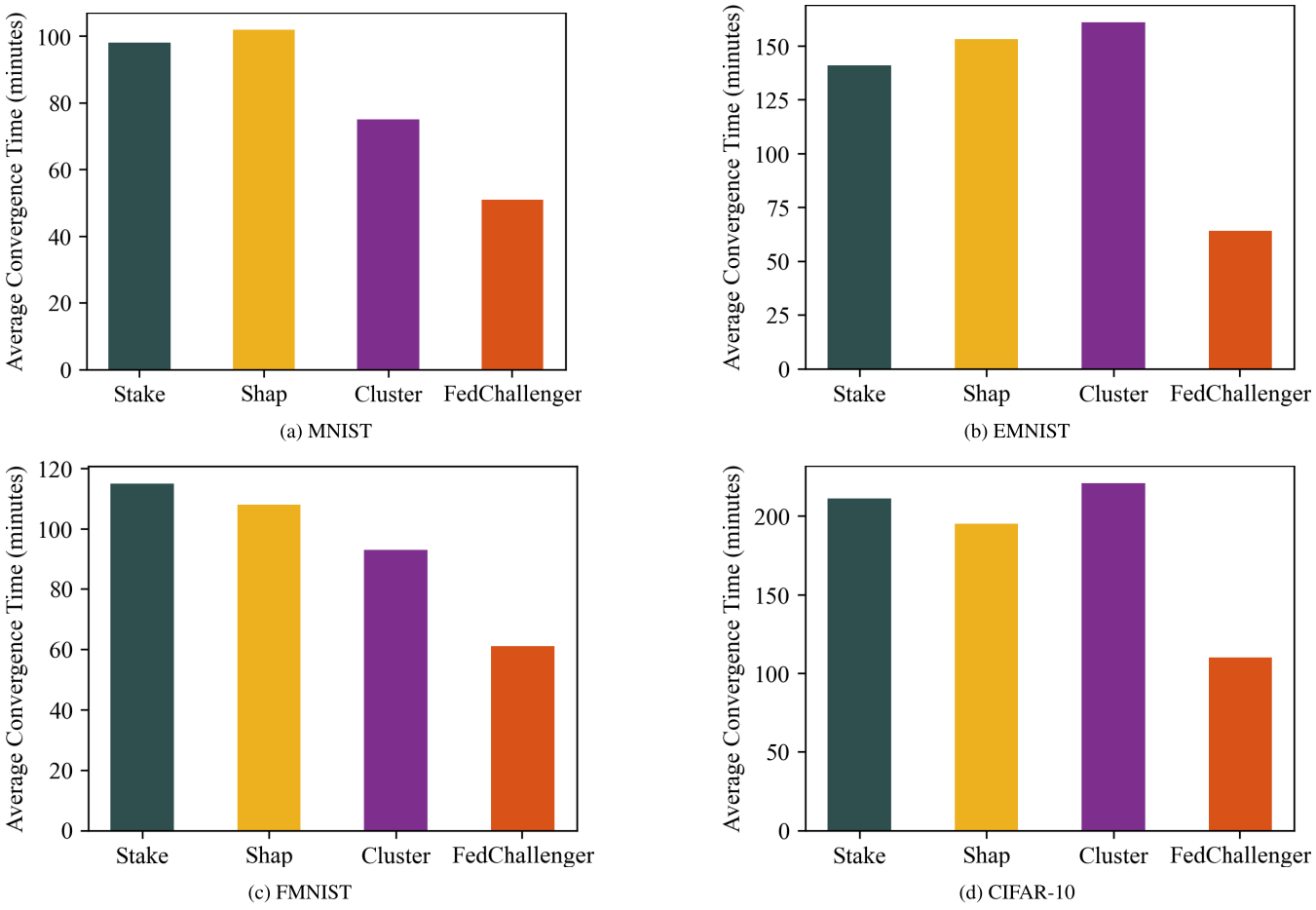


FIGURE 7. No attack scenario: Evaluation of average convergence time.

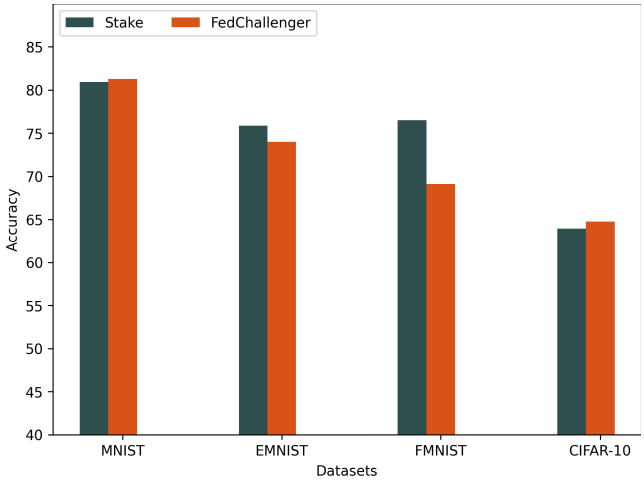


FIGURE 8. Model poisoning attack scenario: Evaluation of accuracy under 40% compromised device scenario.

Shap shows approximately a 1.5% drop in accuracy compared to the FedChallenger in the same compromised device scenario. The stake technique is the worst performer in a 40% compromised device scenario since it loses over 10% accuracy to FedChallenger. However, in FMNIST evaluation at 40% compromised device scenario, the Cluster technique becomes the worst performer with a drop of nearly 20% of accuracy compared to the FedChallenger, while Stake and Shap lose 9.5% and 6% accuracy to Fedchallenger.

TABLE 2. Accuracy comparison with 30% label-flipping on MNIST.

Method	Accuracy (%)
FedAvg	48.2 ± 1.4
Krum	53.7 ± 2.1
Trimmed-Mean	83.5 ± 1.8
FedChallenger	$\mathbf{86.7 \pm 0.9}$

Finally, in CIFAR-10 evaluation, most techniques start to lose more accuracy compared to FedChallenger due to the nature of the dataset, and Shap becomes a bad performer in 40% compromised device scenario as it loses nearly 15% accuracy to the FedChallenger. Additionally, the Stake and Cluster technique loses 10% and 12% global model accuracy, respectively, compared to FedChallenger under the same compromised device percentage.

From the evaluation, it is evident that the FedChallenger, due to its challenge-response mechanism and consideration of the variant of Trimmed-Mean aggregation that employs MAD-based adaptive m value pruning with cosine-similarity computation, competes significantly well with Stake, Shap,

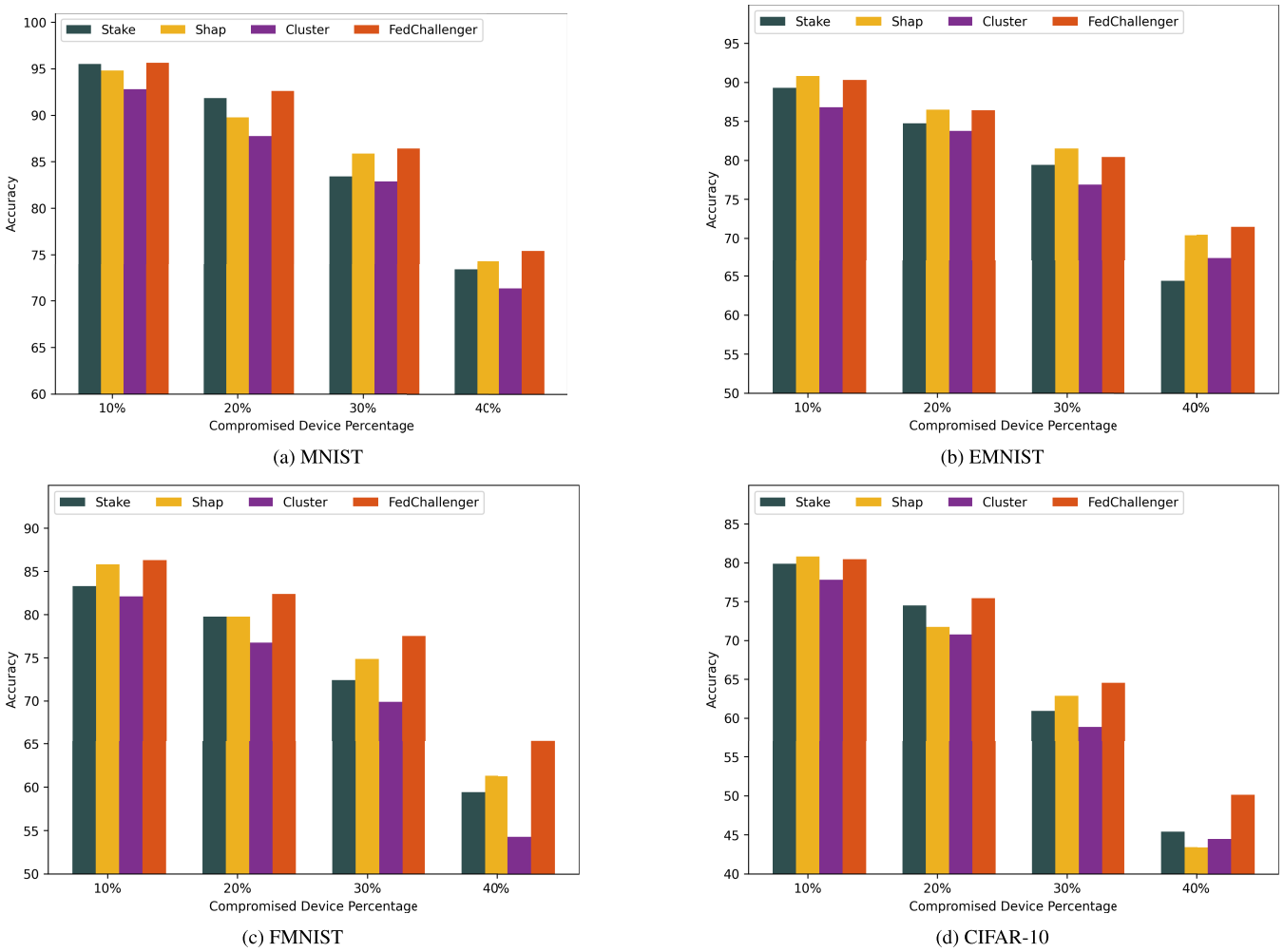


FIGURE 9. Data poisoning attack: Evaluation of accuracy under different compromised device percentage.

TABLE 3. F1-Score for techniques under different percentage of compromised devices.

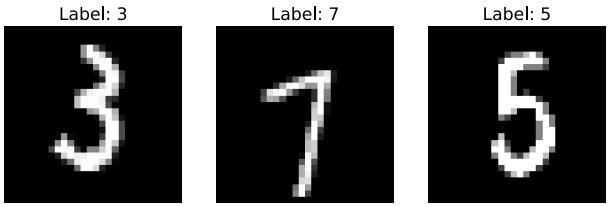
Techniques	$\alpha = 0$		$\alpha = 10$		$\alpha = 20$		$\alpha = 30$	
	EMNIST	CIFAR-10	EMNIST	CIFAR-10	EMNIST	CIFAR-10	EMNIST	CIFAR-10
Trimmed-Mean	91.34	83.93	87.56	76.35	82.54	68.55	74.52	60.37
Krum	89.25	78.67	80.34	67.54	65.81	51.46	48.24	34.22
FedAvg	91.84	83.34	72.33	57.38	53.17	38.48	34.35	23.46
DUEL	90.26	81.43	78.53	78.39	82.47	71.54	74.37	61.33
Stake	91.94	82.94	89.39	79.86	84.75	74.50	79.47	63.91
Shap	91.19	**84.14**	90.86	80.83	86.53	73.75	**81.58**	64.87
Cluster	90.54	81.54	86.85	77.83	83.75	70.75	76.87	60.87
FedChallenger	**92.48**	83.98	**91.37**	**81.45**	**86.74**	**75.45**	80.4	**65.53**

and Cluster techniques. Once again, it has been proven that the method is robust across diverse datasets and converges faster due to the lightweight nature of its algorithms.

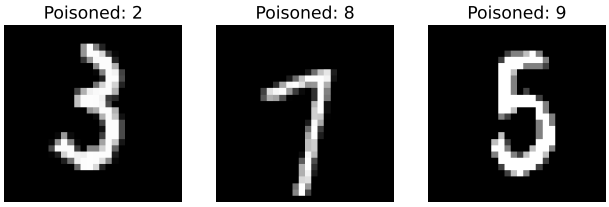
E. DEMONSTRATION WITH EXEMPLARY MNIST DATA

To empirically validate the FedChallenger approach, we demonstrate its behaviour on a subset of the MNIST dataset under label-flipping attacks. Figure 10 shows the original and poisoned samples along with FedChallenger's mitigation process. In original samples, 3, 7, and 5 are the correct labels for the images. However, the poisoned samples are identified with incorrect labels of 2, 8, and 9, respectively. After FedChallenger's corrections, labels 3, 7, and 5 are restored. Table 2 quantifies the impact of 30% label-flipping on MNIST classification:

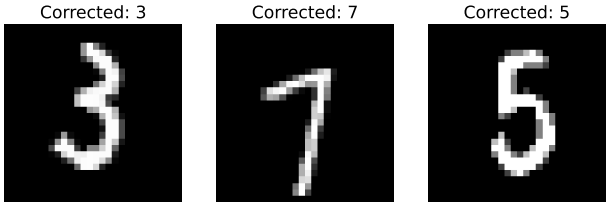
The demonstration shows FedChallenger maintains 86.7% accuracy despite label-flipping, outperforming baseline methods by 4-57%. Figure 11 illustrates how cosine similarity detects malicious updates from poisoned clients. From the figure, it is evident that FedChallenger successfully identifies poisoned updates with similarity < 0.3 while retaining benign ones with similarity > 0.7.



(a) Original samples (labels: 3, 7, 5)



(b) Poisoned samples (flipped to: 2, 8, 9)



(c) FedChallenger corrections

FIGURE 10. Demonstration of label-flipping attack and mitigation on MNIST samples.

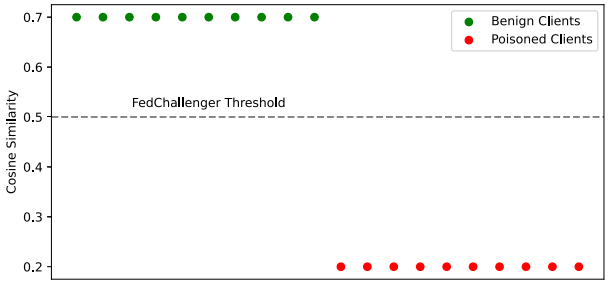


FIGURE 11. Gradient cosine similarity between benign and poisoned clients.

F. ABLATION STUDY

To further analyze the impact of adversarial settings and hyperparameters, we conduct an ablation study.

1) IMPACT OF COMPROMISED DEVICE PERCENTAGES

Table 3 presents an ablation study to understand better how state-of-the-art techniques perform under varying percentages of compromised devices with poisoned data. The study is conducted under the following experimental setup:

- **Datasets**: Two widely used datasets, EMNIST and CIFAR-10, are employed to evaluate the techniques.
- **Adversarial Conditions**: The percentage of compromised devices α is varied at 0%, 10%, 20%, and 30% to simulate increasing levels of adversarial influence.
- **Metric**: The F1-Score is used to measure the performance of each technique.

The findings of the ablation study are highlighted as follows:

TABLE 4. F1-Scores (%) under varying B and η.

Techniques	$B = 10$ $\eta = 0.001$	$B = 20$ $\eta = 0.001$	$B = 20$ $\eta = 0.01$
FedAvg	36.54	38.48	40.69
Krum	48.37	51.46	55.64
FedChallenger	**83.15**	**75.45**	**80.11**

- **Baseline Performance** ($\alpha = 0$): All techniques perform well without adversarial influence. FedChallenger achieves the highest F1-Score of 92.48% on the EMNIST dataset, while the Shap technique achieves the highest F1-Score of 84.14% on CIFAR-10. Stake remains a close contender on EMNIST with an F1-Score of 91.94%, and FedChallenger ranks second on CIFAR-10 with an F1-Score of 83.98%.
- **Impact of Adversarial Conditions** ($\alpha > 0$): As the percentage of compromised devices increases, the performance of most techniques degrades. FedChallenger maintains the highest F1-Score across most levels of α, demonstrating its robustness. However, it achieves the second-highest F1-Score on the EMNIST dataset when 30% of devices are compromised, with Shap emerging as the clear winner. In contrast, FedAvg shows significant vulnerability, with its F1-Score dropping sharply from 83.34% at $\alpha = 0$ to 23.46% at $\alpha = 30$ on the CIFAR-10 dataset. Krum is the second most vulnerable strategy, with F1-Scores of 48.24% and 34.22% in the 30% compromised device scenario on the EMNIST and CIFAR-10 datasets, respectively.
- **Comparative Analysis**: FedChallenger consistently outperforms other techniques, highlighting its effectiveness in adversarial conditions. Techniques like Stake, Shap, and Cluster exhibit moderate robustness, while FedAvg and Krum are the most vulnerable to adversarial influence.

2) IMPACT OF BATCH SIZE AND LEARNING RATE

This experiment evaluates the effects of batch size (B) and learning rate (η) variations exclusively through F1-Score measurements on the CIFAR-10 dataset, with a fixed $\alpha = 20$. The outcomes are presented in Table 4 where the

Key Findings are:

- **Batch Size**: Larger batches ($B = 20$) generally improve robustness (e.g., FedAvg gains +1.94% over $B = 10$).
- **Learning Rate**: Increasing η to 0.01 boosts performance for FedAvg (+5.75%) but may destabilize Krum.
- **FedChallenger** remains robust across configurations, though $B = 10$ surprisingly outperforms at $\eta = 0.001$.

This study demonstrates the importance of robust aggregation techniques in FL, particularly in adversarial settings. FedChallenger is the most effective method for maintaining high performance even under significant adversarial influence. It is a strong candidate for real-world applications where device data poisoning is a concern. The default setup

($B = 20$, $\eta = 0.001$) balances stability and performance, but tuning η could further mitigate poisoning effects.

VI. CONCLUSION

Existing defence strategies against model poisoning and data poisoning attacks have been shown to inadequately prevent the propagation of malicious samples into the aggregation process. To address this limitation, *FedChallenger*, a zero-trust challenge-response-based defence mechanism, is introduced. This approach accumulates device behavioural data to detect compromised data or model parameters. Additionally, a secondary defence layer uses an adaptive Trimmed-Mean approach in two distinct modes. It applies cosine similarity-based consensus boosting during normal operation to refine model updates. When attacks are detected, it filters compromised data through *MAD* computations. The evaluation results suggest that the *MAD*-based aggregation strategy is very effective in removing the impact of malicious updates. Moreover, based on the evaluation of multiple datasets, including MNIST, EMNIST, FMNIST, and CIFAR-10, it is evident that FedChallenger shows consistent performance and is evident to be the best dataset-independent technique compared to the existing state-of-the-art approaches. The experimental evaluation demonstrates FedChallenger's consistent superiority across multiple attack scenarios and datasets. Under 30% label-flipping attacks on FMNIST, FedChallenger achieves significant accuracy improvements of 8.2%, 47.8%, 88.4%, and 7.4% over Trimmed-Mean, Krum, FedAvg, and DUEL approach, respectively. The advantages extend to convergence speed, with FedChallenger showing 60%, 52%, 31%, and 27% faster performance than these same baselines in 20% label flipping scenarios. For more severe 40% model poisoning attacks on CIFAR-10, FedChallenger maintains substantial accuracy leads, outperforming comparative methods by factors of 1.24 to 6.7 times. Even in benign environments, FedChallenger preserves its competitive edge due to consideration of consensus-boosted gradients. Moreover, the evaluation with recent techniques suggests that FedChallenger is consistently a good competitor of the Stake, Shap, and Cluster techniques and offers 36%, 47%, and 54% faster convergence guarantee in the CIFAR-10 dataset, respectively. These comprehensive results validate FedChallenger as a robust, dataset-independent solution that effectively addresses the limitations of existing defence mechanisms. The technique's multi-layered architecture combines challenge-response verification with enhanced cosine similarity-based aggregation and proves particularly effective at neutralizing diverse poisoning threats while maintaining model performance.

FedChallenger provides robust defence against poisoning attacks but exhibits two limitations: vulnerability to sophisticated privacy attacks (e.g., HidAttack, membership inference) and untested performance on textual data. Future extensions will investigate textual domain adaptation and develop countermeasures against different privacy attacks to broaden the framework's protective scope.

REFERENCES

[1] S. Sarkar, "Demystifying decentralized storage—A critical building block of future of Internet or Web 3.0," *Telecom Bus. Rev.*, vol. 17, no. 1, p. 18, 2024.

[2] M. A. Albreem, A. M. Sheikh, M. H. Alsharif, M. Jusoh, and M. N. Mohd Yasin, "Green Internet of Things (GIoT): Applications, practices, awareness, and challenges," *IEEE Access*, vol. 9, pp. 38833–38858, 2021.

[3] R. Xu, N. Baracaldo, and J. Joshi, "Privacy-preserving machine learning: Methods, challenges and directions," 2021, *arXiv:2108.04417*.

[4] G. Xia, J. Chen, C. Yu, and J. Ma, "Poisoning attacks in federated learning: A survey," *IEEE Access*, vol. 11, pp. 10708–10722, 2023.

[5] L. Sun, J. Tian, and G. Muhammad, "FedKC: Personalized federated learning with robustness against model poisoning attacks in the metaverse for consumer health," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 5644–5653, Aug. 2024.

[6] P. Gupta, K. Yadav, B. B. Gupta, M. Alazab, and T. R. Gadekallu, "A novel data poisoning attack in federated learning based on inverted loss function," *Comput. Secur.*, vol. 130, Jul. 2023, Art. no. 103270.

[7] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, Jan. 2016, pp. 1273–1282.

[8] P. Blanchard, E. M. E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 118–128.

[9] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2018, pp. 5650–5659.

[10] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to Byzantine-robust federated learning," in *Proc. 29th USENIX Conf. Secur. Symp.*, Jan. 2019, pp. 1623–1640.

[11] Y. Zhao, J. Chen, J. Zhang, D. Wu, J. Teng, and S. Yu, "PDGAN: A novel poisoning defense method in federated learning using generative adversarial network," in *Proc. 19th Int. Conf. Algorithms Archit. Parallel Process.*, Jan. 2020, pp. 595–609.

[12] S. Andreina, G. A. Marson, H. Möllering, and G. Karame, "BaFFLe: Backdoor detection via feedback-based federated learning," in *Proc. IEEE 41st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2021, pp. 852–863.

[13] N. Dong, Z. Wang, J. Sun, M. Kampffmeyer, W. Knottenbelt, and E. Xing, "Defending against poisoning attacks in federated learning with blockchain," *IEEE Trans. Artif. Intell.*, vol. 5, no. 7, pp. 3743–3756, Jul. 2024.

[14] D.-P. Khuu, M. Sober, D. Kaaser, M. Fischer, and S. Schulte, "Data poisoning detection in federated learning," in *Proc. 39th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2024, pp. 1549–1558.

[15] N. Moharram Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Defending against the label-flipping attack in federated learning," 2022, *arXiv:2207.01982*.

[16] J. Ye, L. Shi, H. Xu, S. Pan, and J. Xu, "A client detection and parameter correction algorithm for clustering defense in clustered federated learning," in *Proc. 30th Annu. Int. Conf. Mobile Comput. Netw.*, Dec. 2024, pp. 2383–2388.

[17] M. A. Moyeen, K. Kaur, A. Agarwal, S. R. Manzano, M. Zaman, and N. Goel, "FedChallenger: Challenge-response-based defence for federated learning against Byzantine attacks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2023, pp. 3843–3848.

[18] S. I. Popoola, A. L. Imoize, M. Hammoudeh, B. Adebisi, O. Jogunola, and A. M. Aibinu, "Federated deep learning for intrusion detection in consumer-centric Internet of Things," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1610–1622, Feb. 2024.

[19] D. C. Howell, "Median absolute deviation," in *Encyclopedia of Statistics in Behavioral Science*. Hoboken, NJ, USA: Wiley, 2005.

[20] M. Xu, Y. Zeng, M. Xue, J.-L. Zhang, J. Wan, M. Zhou, Y. Wen, and Y. Shi, "FedAG: A federated learning method based on data importance weighted aggregation," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2023, pp. 1–6.

[21] E. Mahdi El Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," 2018, *arXiv:1802.07927*.

[22] Q. Xia, Z. Tao, Z. Hao, and Q. Li, "FABA: An algorithm for fast aggregation against Byzantine attacks in distributed neural networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 4824–4830.

[23] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4574–4588, 2021.

[24] M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar, "The security of machine learning," *Mach. Learn.*, vol. 81, no. 2, pp. 121–148, May 2010.
[25] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
[26] O. Suciu, R. Marginean, Y. Kaya, H. Daumé, and T. Dumitras, "When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks," in *Proc. 27th USENIX Secur. Symp. USENIX Secur.)*, Mar. 2018, pp. 1299–1316.
[27] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 19–35.
[28] Y. Chen, F. Luo, T. Li, T. Xiang, Z. Liu, and J. Li, "A training-integrity privacy-preserving federated learning scheme with trusted execution environment," *Inf. Sci.*, vol. 522, pp. 69–79, Jun. 2020.
[29] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
[30] Z. Gu and Y. Yang, "Detecting malicious model updates from federated learning on conditional variational autoencoder," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2021, pp. 671–680.
[31] M. Arafeh, A. Hammoud, H. Otrok, A. Mourad, C. Talhi, and Z. Dziong, "Independent and identically distributed (IID) data assessment in federated learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2022, pp. 293–298.
[32] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.
[33] G. Zizzo, A. Rawat, M. Sinn, and B. Buesser, "FAT: Federated adversarial training," 2020, *arXiv:2012.01791*.
[34] J. Jithish, B. Alangot, N. Mahalingam, and K. S. Yeo, "Distributed anomaly detection in smart grids: A federated learning-based approach," *IEEE Access*, vol. 11, pp. 7157–7179, 2023.
[35] N. Wang, Y. Xiao, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "FLARE: Defending federated learning against model poisoning attacks via latent space representations," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2022, pp. 946–958.
[36] D. D. S. Braga, M. Niemann, B. Hellingrath, and F. B. D. L. Neto, "Survey on computational trust and reputation models," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–40, Sep. 2019.
[37] X. Zhou, X. Chen, S. Liu, X. Fan, Q. Sun, L. Chen, M. Qiu, and T. Xiang, "FLGuardian: Defending against model poisoning attacks via fine-grained detection in federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 5396–5410, 2025.
[38] X. Chen, C. Feng, and S. Wang, "AIDFL: An information-driven anomaly detector for data poisoning in decentralized federated learning," *IEEE Access*, vol. 13, pp. 50017–50031, 2025.
[39] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.
[40] L. Yang, Y. Miao, Z. Liu, Z. Liu, X. Li, D. Kuang, H. Li, and R. H. Deng, "Enhanced model poisoning attack and multi-strategy defense in federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 3877–3892, 2025.
[41] Y. Ali, K. H. Han, A. Majeed, J. S. Lim, and S. O. Hwang, "An optimal two-step approach for defense against poisoning attacks in federated learning," *IEEE Access*, vol. 13, pp. 60108–60121, 2025.
[42] Y. Ren, Z. Yang, G. Feng, and X. Zhang, "PurifyFL: Non-interactive privacy-preserving federated learning against poisoning attacks based on single server," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 9, no. 3, pp. 2232–2243, Jun. 2025.
[43] P. Tang, X. Zhu, W. Qiu, Z. Huang, Z. Mu, and S. Li, "FLAD: Byzantine-robust federated learning based on gradient feature anomaly detection," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 4, pp. 3993–4009, Jul. 2025.
[44] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Proc. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Jan. 2017, pp. 409–437.
[45] Y. Long, Z. Xue, L. Chu, T. Zhang, J. Wu, Y. Zang, and J. Du, "FedCD: A classifier debiased federated learning framework for non-IID data," in *Proc. 31st ACM Int. Conf. Multimedia*, Oct. 2023, pp. 8994–9002.
[46] H. Xie, Y. Wang, Y. Ding, C. Yang, H. Zheng, and B. Qin, "Verifiable federated learning with privacy-preserving data aggregation for consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2696–2707, Feb. 2024.
[47] A. Koloskova, S. U. Stich, and M. Jaggi, "Sharper convergence guarantees for asynchronous SGD for distributed and federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 17202–17215.
[48] H. Brendan McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," 2017, *arXiv:1710.06963*.
[49] M. Yang, H. Cheng, F. Chen, X. Liu, M. Wang, and X. Li, "Model poisoning attack in differential privacy-based federated learning," *Inf. Sci.*, vol. 630, pp. 158–172, Jun. 2023.
[50] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, "Ethereum smart contract analysis tools: A systematic review," *IEEE Access*, vol. 10, pp. 57037–57062, 2022.

M. A. MOYEEN (Graduate Student Member, IEEE) received the B.Sc. degree in computer science and engineering from Khulna University of Engineering and Technology, Bangladesh, and the M.Sc. degree in computer science from Dalhousie University, Halifax, Canada. He is currently pursuing the Ph.D. degree with Concordia University, Montreal, Canada. His notable research contributions fall mainly under federated learning, software-defined networks, and the network security domain.

KULJEET KAUR has been an Associate Professor at the École de Technologie Supérieure (ÉTS), Montreal, since 2020. She published over 75 scientific/technical articles and three books. She has secured research funding from various sources, such as the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds de Recherche du Québec–Nature et Technologies (FRQNT), the Department of Science and Technology (DST), and TCS Innovations Laboratories. Her research interests include cybersecurity, cloud/edge computing, the Internet of Things (IoT), applied machine learning and artificial intelligence, communications, and smart grids.

Dr. Kaur served as a Technical Program Committee (TPC) Member for several international conferences, including IEEE GLOBECOM and IEEE ICC. She was a recipient of the 2023 N2Women Rising Stars in Networking and Communications and the 2021 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence in Scalable Computing for Early Career Researchers. She was also awarded the 2021 IEEE System Journal and the 2018 IEEE ICC Best Paper Awards. She also received the Best Research Paper Awards from the Thapar Institute of Engineering and Technology, in 2022 and 2019. She has been serving as an Editor/Guest Editor for various international journals of repute, such as *Computer Communications* (Elsevier), *Ad Hoc Networks* (Elsevier), *Security and Privacy* (Wiley), *Journal of Information Processing Systems*, *Human-Centric Computing and Information Sciences* (Springer), *Frontiers in Communications and Networking*: Board of Smart Grid Communications, and *International Journal of Applied Engineering Research* (IJAER). She has organized different special issues at different venues, such as IEEE Internet of Things Journal, IEEE Transactions on Industrial Informatics, IEEE Transactions on Consumer Electronics, and IEEE Open Journal of the Computer Society. She has been organizing international symposiums and workshops of several flagship conferences, such as IEEE GLOBECOM, IEEE INFOCOM, IEEE ICC, ACM MOBICOM, and others. She serves as the Faculty Representative for the IEEE ÉTS Student Branch and a Secretary for the IEEE ComSoc Women in Communications Engineering (WICE). She served as the Deputy Secretary for the IEEE WICE, from 2022 to 2024, the Vice Chair for the IEEE Montreal Young Professionals Affinity Group, from 2020 to 2022, and the Website Co-Chair for the Networking Women (N2Women), from 2020 to 2022, a discipline-specific community for female researchers.

ANJALI AGARWAL (Senior Member, IEEE) received the B.E. degree in electronics and communication engineering from Delhi College of Engineering, India, the M.Sc. degree in electrical engineering from the University of Calgary, Alberta, and the Ph.D. degree in electrical engineering from Concordia University, Montreal. She is currently a Professor with the Department of Electrical and Computer Engineering, Concordia University, Montreal. Prior to joining faculty in Concordia, she was a Lecturer with IIT Roorkee and a Protocol Design Engineer and a Software Engineer in industry. She has worked in various aspects of wireless networks. Her current research interests include cloud computing, security, resource management, fault management, and energy management.

S. RICARDO MANZANO received the M.Sc. degree in electrical and computer engineering from the University of Waterloo, in 2020. He is currently a Research Scientist at Cistech Ltd., and Cistel Technologies, Ottawa, Canada. He is a Notable Researcher in the areas of cybersecurity, cloud, networking, the IoT devices, and artificial intelligence. He is with Cistel Technologies, working on government projects and continuing his research as a Scientist. He has extensive industrial experience in robotics, networking, and AI. He began his career at Telefónica Ecuador, a multinational telecommunications company based in Spain, where he was a Pre-Sales Engineer in network solutions, including LAN, WAN, cloud, and security. Following this, he joined the University of Waterloo as a Research Assistant. His research interests include artificial intelligence, security, networking, and robotics.

MARZIA ZAMAN received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Memorial University of Newfoundland, Canada, in 1993 and 1996, respectively. She began her career at the Software Engineering Analysis Laboratory, Nortel Networks, Ottawa, ON, Canada, in 1996, and later joined the OPTera Packet Core Project as a Software Developer. With extensive industry experience, she was a Researcher and a Software Designer at Accelight Networks, Excelocity, Sanstream Technology, and Cistel Technology. She has collaborated with researchers from numerous Canadian universities on various industry-academic research and development projects. Her research interests include renewable energy, wireless communication and networks, machine learning, deep learning, and software engineering.

NISHITH GOEL received the Bachelor of Engineering degree in Jodhpur, India, and the M.A.Sc. degree in electrical engineering and the Ph.D. degree in systems design engineering from the University of Waterloo. He began his professional career at Bell-Northern Research, Ottawa, in 1984, and then he moved to Northern Telecom, in 1988. He is currently the CEO of Cistel Technology, an information technology company he founded, in 1995, which has operations in Canada and USA. He is the Co-Founder of CHiL Semiconductor, iPine Networks, and Sparq Systems. He is also the Chair of the Queen's Center for Energy and Power Electronics Research, Queen's University. He serves on various corporate boards of directors. His research interests include information technology, especially in the areas of wireless networks, cloud computing, and network security.