ChallengeShift-FL: Rotating Behavioral Fault Challenges for Detecting Stealthy Poisoning in Non-IID Federated Industrial Learning

Sayor Ajfar Aaron[*], Md. Hasanuzzaman[**], Sajjat Hossain Abir[**], Ashif Newaz[**], Md. Mushfiqur Rahman[**], Md Monir Hossain[**],

[*] ORCID: https://orcid.org/0009-0002-5381-7738
Department of Computer Science and Engineering, Canadian University of Bangladesh
[**] ORCID: https://orcid.org/0009-0002-2868-8743
Department of Computer Science and Engineering, Canadian University of Bangladesh
[**] ORCID: https://orcid.org/0009-0008-7500-5689
Department of Computer Science and Engineering, Canadian University of Bangladesh
[**] ORCID: https://orcid.org/0009-0005-6445-3011
Department of Computer Science and Engineering, Canadian University of Bangladesh
[**] ORCID: https://orcid.org/0009-0006-8483-8653
Department of Computer Science and Engineering, Canadian University of Bangladesh
[**] ORCID: https://orcid.org/0009-0006-8922-0297
Department of Computer Science and Engineering, Canadian University of Bangladesh

DOI: 10.29322/IJSRP.16.10.2026.p17811
https://dx.doi.org/10.29322/IJSRP.16.10.2026.p17811

Paper Received Date: 12th August 2026
Paper Acceptance Date: 11th October 2026
Paper Publication Date: 28th October 2026

Abstract- We investigate whether rotating controlled fault challenges improves behavioral screening of poisoned client updates under non-IID federated learning. ChallengeShift-FL compares candidate models with the current global model on sensor-window interventions, applies a normal-data stability allowance, and maintains bounded temporal trust for soft aggregation. A reproducible synthetic experiment executes seven methods across seven scenarios and three paired seeds: 147 runs and 3,528 federated rounds. Comparisons include FedAvg, coordinatewise median, fixed and rotating behavioral weighting, the complete design, and two ablations. Rotation does not consistently reduce target-fault suppression. Under limited-knowledge adaptive poisoning, instantaneous rotation changes target-to-normal error from 54.22% to 53.78%, a paired difference of −0.44 percentage points; the complete design performs worse than its no-rotation counterpart. Under scaled replacement, the complete design yields 99.72% target-to-normal error, compared with 40.11% for instantaneous rotating weighting and 21.22% for coordinatewise median. Its stability allowance reduces honest suspicious flags but also admits damaging fault-suppression updates. All candidates retain positive aggregation weights, so zero hard rejection is structural rather than evidence of fairness. The findings expose failure modes of normal-only tolerance and trust inertia. They do not establish an industrial security guarantee or a general advantage of rotation. Challenge coverage and sample refresh remain confounded, and validation on public physical fault datasets is required.

Index Terms- Adaptive poisoning, behavioral validation, federated learning, industrial fault classification, non-IID data, rotating challenges.

I. INTRODUCTION

The Failure to interest in the findings of this research relates to selective fault blindness: a common classifier identifies multiple types of faults (including normal operation) as well as various types of faults; however, it incorrectly assigns one specific type of fault into the "normal" category. The authors examine this failure in a federated network of simulated industrial plants. The plant locations send their respective models to each other instead of sending sensor readings. An adversarial location can take advantage of how much influence they have over the global classifier by providing an abnormal parameter update. Similarly, an honest plant location may provide an abnormal parameter update since the operational modes and frequency distributions of fault classes for both its own and neighboring plants differ. Thus, the server has a discrimination problem. It needs to determine when a malicious update is being made without incorrectly assuming that an update that differs from others submitted by clients is evidence of malicious activity.

To address this discrimination issue, ChallengeShift-FL proposes that the server maintain a set of controlled sensor window intervention with labeled classes. Additionally, the server will modify the challenge conditions used to assess new updates. Each submission is treated as a slight modification to the currently global model. The server evaluates degradation on fault

This publication is licensed under Creative Commons Attribution CC BY.

challenges, determines whether an apparent anomaly in behavior is consistent with ordinary benign variation, and calculates a bounded trust measure for each client based on client behavior. Client trust affects the weighting of the client's contribution. To be transparent, the system was designed so that everything is visible including the data generator, local optimizer, attacker, validation score, and aggregation function.

Our research question is whether modifying the challenge conditions results in increased resistance to an adaptive poisoning adversary compared to a fixed challenge condition with the same overall architecture, while minimizing penalties against honest clients due to non IID client behavior. Our question allows for a negative result. Although rotation can allow for assessment across additional conditions, there exists the possibility that all conditions contain a learned vulnerability. Rotation can also punish legitimate specialization in a single fault regime. Therefore, we record both successful attacks and penalties against honest clients and compare matched versions of our experiments, excluding overall aggregate classification accuracy as a proxy for either.

The study contributes three aspects related to design and experimental methodology. First, we describe a method to rotate fault-oriented behavioral assessments as part of the server side functionality which is inspectable. Second, we link that assessment to a benign stability adjustment and a bounded temporal trust estimation. Finally, we implement two sets of comparative experiments to demonstrate separately the effectiveness of rotating challenges, trust memory, and stability adjustments.

The above contributions relate to design and experimental methodologies. As such, their practical applicability is limited to the described simulation scenarios below; no claims were made regarding novelty or universality of security.

II. RESEARCH MOTIVATION AND TESTABLE EXPECTATIONS

A fixed bank has an attractive operational property: the server observes candidate updates on the same examples in successive rounds. Score changes then have a stable interpretation. The cost is reuse. An attacker who knows some bank examples can preserve predictions on those examples while changing behavior elsewhere. Rotating conditions remove this particular reuse opportunity only when the new conditions probe behavior that the attacker has not already preserved. Randomness alone is not enough. A bank consisting of many copies of the same discriminative direction may provide little additional coverage.

The contrast tested here is therefore behavioral coverage rather than secrecy as an absolute security boundary. Limited attacker knowledge covers a small known bank; extensive knowledge covers more challenge families but not the server's current random draws. Neither setting grants access to final evaluation data. An attacker trained against the challenge distribution is still adaptive even when it lacks the exact current examples. If rotation succeeds only with limited knowledge, the appropriate conclusion is that coverage uncertainty helped under that knowledge model. If the effect disappears with extensive knowledge, that disappearance is itself evidence against a general defense claim.

We state the primary expectation as a directional hypothesis: under adaptive limited knowledge, rotation reduces the fraction of held-out targeted fault windows classified as normal relative to fixed verification. The corresponding falsification condition is an equal or higher missed-target fraction under rotation, or a benefit accompanied by an unacceptable increase in honest penalties. A second expectation is that benign-stability protection lowers the rate at which honest updates receive low weight under severe heterogeneity. A third is that trust memory smooths intermittent judgments without retaining enough stale trust to permit persistent suppression. The measurements evaluate these expectations separately; none is assumed by the architecture.

The experimental unit for uncertainty is an independently initialized simulation seed. Clients and rounds within one simulation are correlated observations because they share the global model. The reported standard deviations therefore summarize between-seed variability, not independent evidence from every client-round pair. With only a small number of repetitions, the study emphasizes effect sizes and failure patterns. It does not infer statistical significance from narrow-looking plots or claim that a mean difference estimates a population effect in industrial deployments.

III. RELATED WORK AND RESEARCH GAP

a) Federated industrial diagnosis and the heterogeneity problem

The motivation for this work is the ability of a federated industrial diagnosis server to make decisions regarding whether a local model is usable for a common task given that each device has varying levels of expertise. The following studies provide the necessary background to define the design space for federated diagnosis.

Federated Averaging, which aggregates local updates of a model across multiple devices, while keeping all local training data on the respective devices, was introduced in [1]. Kairouz et al. identify the primary areas of concern (optimization, privacy, security, and heterogeneous systems) in [4], related to Federated Learning. Non IID federated learning are reviewed and treated as a separate optimization problem in [17], which requires explicit modeling and mitigation. A review of federated diagnosis for Condition Monitoring (CM) provides an overview of CM, federated diagnostics, Data Island constraints and Industrial Privacy/Security concerns [38]. Qian et al., describe federated transfer diagnosis via private federated learning, federated communication, and distribution alignment mechanisms [16]. Zhang et al., present a method for Machinery Fault Diagnosis via Self-Supervised Training and Dynamic Validation, excluding weaker performing local models from aggregation [15]. A method for intra-client Imbalance Driven Fault Diagnosis utilizes class specific update gains to determine how much of a local model's contribution should be included in federated aggregation on a small balanced dataset [25]. FedFZ uses Fuzzy-Rule Fusion at the Layer level and conducts Chemical Process Fault Diagnosis utilizing the Tennessee Eastman Process [26]. FedSC creates Personalized Diagnostic Models and determines how these models should be weighted when aggregated based on a similarity mechanism with Local Distance Constraints [34]. A Distributed Bearing-Diagnosis Method utilizes Domain

This publication is licensed under Creative Commons Attribution CC BY.

Adversarial Learning, Maximum Mean Discrepancy and Discrepancy Weighted Parameter Aggregation [37]. FedDPA addresses drift due to Non IID Client Data by utilizing Client Class Prototypes, Hierarchical Prototype Aggregation and Contrastive Alignment [23]. A Quantity Skew Aware Method Estimates the Dirichlet Parameter for Regularization and Gradient Interference Adjustment to account for Non IID Training Data [40]. Class Imbalance Mitigation combines Within-Client Balancing, Client Selection, Sample Size Adjustment and Adaptive Local Training Parameters [41]. AS-CFL Dynamically Forms Client Clusters Based on Model Update Similarity and Utilizes a Forward-Incentive Mechanism [42]. AFLS Selects Parallel or Sequential Federation Procedures based on the Estimated Degree of Non IID Data [43]. DAPA Estimates Distribution Sketches, Derives Wasserstein Affinity and Establishes Hierarchical Aggregation Weights for Each Heterogeneous Client [44].

Given the mechanisms above, there is a Design Constraint for Challenge Shift FL; A Server Cannot Equate Weak Performance on Unrepresented Fault Families with Malicious Intent. Therefore, the Proposal Calibrates the Suspicious Update Penalty Relative to Concurrent Honest Variation and Preserves Minimum Aggregation Weight. This is a Practical Protection to be Measured, Rather Than a Proof That Every Honest Operating Regime Remains Equally Represented.

b) Industrial diagnostic representation, sparse faults, and transfer

The introduction of Fault Diagnosis adds a further distinction between missing knowledge and intentionally suppressed knowledge. We treat this distinction operationally within our experimental design using an honest client stability control.

Diagnostic representative methods solve the representation and transfer problems as follows:

A federated transfer diagnostic approach that integrates Variational Auto-Encoding (VAE) and Few-Shot Learning with Regularized Averaging addresses cross-unit diagnostic challenges. [24] A federated peer-to-peer multi-model transfer learning method that integrates Multitask Model Transfer with Adaptive Fusion, provides a solution to Wind Turbine Cluster Fault Diagnostics. [27] A federated Convolutional Denoising Representations and Metric Learning Method (FedCDAE-MN), addresses federated few-shot diagnostics between different Operating Conditions and Fault Types. [28] FedAnchor separates a Shared Model Core from Adaptable Extensions to provide fault diagnostics across Clients with varying Resource Budgets. [29] An Asymmetric Long-Tail Mechanical Diagnostic Approach, utilizes Attention, Asymmetric Convolution, and Margin Calibration to represent Rare Fault Categories more effectively. [30] Relation-Guided Federated Compression utilizes Weighting based on Rare-Fault-Awareness and Constrained Model Slimming to retain Diagnostic Knowledge in Compact Motor Models. [31] SSFL-ATT uses Attention-Based Filtering to remove Unreliable Information Transferred from Unlabeled Clients during Semi-Supervised Federated Fault Diagnoses. [32] ProtoFed combines Episodic Prototypical Learning with Global Prototype Calibration to enable few-shot Rolling Bearing Diagnoses under Heterogeneous Operating Conditions. [33] Federated Cross-Robot Diagnostics integrates Individual Fault Models with Group Speed and Direction Consistency to enable System-Level Peer Diagnostics. [35] Repair-Domain Federated Transfer Diagnostics utilizes Random Forest Reconstruction and Domain Discrepancy Losses to address Missing Signal Fragments and Negative Transfer. [36] Self-Attention Clustering for Bearing Diagnoses enables Equipment Clustering by Distribution Similarity prior to Collaborative Training. [39] We do not replicate such diagnostic architectures in our simulation. Rather we isolate a smaller question: will changes to the Validation Distribution affect the leverage of a suppression attacker? By utilizing a simpler model we may be able to identify a mechanism; however, we cannot establish whether the challenge families preserve meaningful physical characteristics in Rotating Machinery, Chemical Plants, etc., Multi-Robot Systems. Thus, the Final Diagnostic Evaluation must differentiate between Intervention Coverage and Physical Fidelity.

c) Aggregation, server reference data, and behavioral inspection

The comparison separates statistical aggregation from evaluation of model behavior. Neither type of defense should be credited with performance against an attacker that was omitted from its own threat model. The following descriptions identify mechanisms, without importing reported benchmark improvements into the present experiment.

Yin et al. analyze coordinate-wise median and trimmed-mean distributed gradient algorithms under explicit Byzantine and statistical assumptions. [2] FLTrust uses a small clean server dataset to establish a reference update, assigns trust from directional agreement, and normalizes client-update magnitudes. [5] FLAME combines model clustering, weight clipping, and calibrated noise injection to mitigate federated backdoors while limiting benign-task degradation. [6] DeepSight characterizes neural-network internal structure and outputs to filter poisoned clusters while explicitly addressing benign clients with divergent data distributions. [7] FLDetector identifies suspicious clients from disagreement between received updates and historical-update predictions across multiple iterations. [8] LoMar scores client updates through neighborhood-relative kernel-density estimation and applies a statistically motivated malicious-update threshold. [9] BaFFLe incorporates client feedback about class-specific model performance when deciding whether a model update is genuine. [10] LeadFL adds a local regularization mechanism intended to complement server defenses against bursty targeted and backdoor attacks. [11] PBA filters aggregation outliers using approximate distances computed from binary sequences and a three-sigma criterion. [45] SemLRA normalizes task losses using a within-round median and median absolute deviation to construct bounded reliability coefficients in semantic federation. [46] An asynchronous IoV aggregation method combines delay-aware grouping with within-group truth estimation and encrypted aggregation. [47] FLRAM combines isolation-forest and density-based filtering with a credibility matrix before robust gradient aggregation. [48] FedLucra uses validation loss and local data volume together with exponential smoothing and interquartile filtering to stabilize aggregation weights. [49] FedCWA generates a proxy dataset from class

This publication is licensed under Creative Commons Attribution CC BY.

prototypes and estimates client credibility from prediction-logit similarities. [50] DARA combines adaptive privacy-noise scaling with direction-aware continuous aggregation weights under Byzantine disturbances. [51] The gamma-mean estimator applies density-power-divergence weighting to reduce the influence of Byzantine client messages. [52] FedAA uses continuous-control reinforcement learning and validation-guided rewards to select aggregation weights for fairness and robustness. [53]

Trust weighting is consequently an established design ingredient, and evaluating candidate behavior is also established. The narrower proposed contribution is to combine family rotation with an explicit benign-stability adjustment and measure whether that combination improves the attack–fairness tradeoff. A coordinate median is retained as a computationally feasible baseline, with no assertion that its published guarantees transfer to arbitrary heterogeneous industrial distributions. Fixed and rotating validation must use equal challenge counts and matched seeds so that the comparison does not confuse more validation data with rotation.

d) Adaptive attack surfaces and evaluation discipline

The attack suite developed for this paper is intentionally limited compared to the overall Federated Threat Landscape. The focus has been on fault-to-normal suppression (i.e., normalizing suppressed faults) and candidate-update behavior; additional architectural designs below are proposed as potential extensions and do not represent operational baseline implementations. Attackers may introduce their preferred misclassification behaviors into a targeted client's model via manipulation of client-side models. Model-replacement-based defenses can be designed to defend against evasion attacks. [3] Wan et al. define three categories of wireless federated backdoors based on their poisoning objectives (target), phases of attack, and stages of defense: poisoning data vs. poisoning models. [18] FLARE examines both sparse index mask patterns and model update signature similarities in order to identify poisoning attacks in federated communication that have been sparsified. [19] DDFed includes fully homomorphic aggregation combined with encryption of similarity assessments and feedback-directed selection to address poisoning attacks even though there is only partial visibility of updates. [20] AdaAggRL uses Maximum Mean Discrepancy to estimate similarity in distributions, and learn adaptive weights for aggregation from current, historical, and global distributions. [21] The FLPoison Benchmark evaluates both data and model poisoning together, and also allows for the variation of federated algorithms and data heterogeneities. [22] KPBAFL restricts the backdoor attack space by targeting only a small portion of key model parameters and utilizes feedback from beacons to modify federated backdoor behaviors. [54] A federated meta-learning defense generates class predictions from feature similarities to labeled example support classes in order to decrease the likelihood of successful shared-model backdoor attacks. [55] Cerberus poisoning simultaneously optimizes collaborating clients and triggers to minimize detectable poisoned-model differences. [56] BapFL analyze and launch attacks on the heterogeneous classifier architecture utilized by parameter decoupling in federated learning for personalized classification tasks. [57] BADFSS launches attacks against federated self-supervised learning methods utilizing supervised contrastive poisoning and attention alignment. [58] Label-free vertical backdoor attacks utilize local embedding gradient information to deduce semantic label information without having to assume direct access to labels. [59] Peer-to-peer federated backdoor attacks use graph structure to choose malicious nodes under limited network visibility constraints. [60]

In general, how one interprets the outcome of an adaptive experiment will depend on exactly which elements of the generator distribution the attacker knows. Knowing the generator distribution does not necessarily equate knowing its current random draw(s), nor does it imply the ability to see future held-out test data. The attacker's known state was declared in advance within the experimental design of this paper, the success of an attack is measured relative to a distinct fault test set, and the random variables associated with challenge sampling were separated from those associated with final evaluation. Therefore, if results favoring rotation are obtained then they provide a basis for making a bounded comparative statement about performance; however, they do not resolve white-box optimizations or colluding attackers.

e) Closest recent industrial defenses and the proposed distinction

Three specific 2026 studies are actual and sufficiently limit the novelty of the claimed invention. Each should be considered as being relatively similar in time (close predecessor), rather than as simply different comparison points.

FedRISC-IIoT includes reputation-based participation, layer-by-layer aggregate validation, and light-weight shadow validation in an Industrial IoT defense loop. [12] FL-TWIN combines digital twin snapshots and rollback, along with layer-by-layer update validation, privacy perturbations, dynamic aggregators and auditable DDoS detection. [13] DBST-FL increases gradient derived behavioral trust by semantic trust evaluated from clean and sensitive-validation examples. [14]

FedRISC-IIoT has been designated for publication in December 2026 but can be discovered at any time prior to the October cut-off via its published records online. Therefore, it's acceptable to include the publisher year in the bibliography but exclude the future issue month as the access date. [12] Both FedRISC-IIoT and ChallengeShift-FL rely upon validation driven reputation and reduced malicious impact. [12]

However, FL-TWIN also relies on history of states of digital twins and recovery, while this research evaluates the effectiveness of a light-weight behaviorally generated random challenge. [13] Additionally, DBST-FL uses semantic validation and non-compensatory fusions of trust in relation to server evaluations. [14] The defensive difference between this research and the other two studies relates to control over rotation of virtually faulty families of challenges based on partial knowledge of attackers and imposition of penalties on honest clients. None of the above comparisons at an abstract level provide evidence that rotating challenges is novel, nor does this research make claims to first-of-its-kind status.

This publication is licensed under Creative Commons Attribution CC BY.

IV. PROBLEM FORMULATION AND THREAT MODEL

Client i has a sample of D_i labeled data and each client has a different amount of sample size and class mixtures which define how much they contribute to the common model during the ordinary training process. In the case of our research, non-identically distributed (Non-IID) behavior is not simply a problem of unknowns, but rather it is part of how we design the generator to distribute labels and regimes among clients. Therefore, all clients will operate in parallel, so that both honest and dishonest candidate may be evaluated under identical round states.

The server is assumed to be trusted in order to maintain its validation generator and current challenge realization private, to score accurately and to hold back the last evaluation of the test data. The server will obtain the complete model updates. There will be no "secure aggregation" layer that conceals individual updates. Our architecture assumes that there exists some way to reveal only the sum of updates. Without some form of additional machinery, we could never expect to implement the per-client verification that we propose. We assume reliable communication and therefore do not simulate packet loss, asynchronous staleness, client authentication failures nor server compromise. Each malicious client controls his/her training labels and submitted parameter values according to the implementation of the various attack routines. Since we are trying to modify a single fault label to indicate "normal", we use two types of attacks. Label flipping is used to alter the local supervision for that fault. Model replacement is used to amplify the modification sent to the aggregation point. The adaptive version of these attacks are able to optimize suppression while maintaining a preservation objective at times when the attacker knows what the challenge family was. Each type of attack is an executed computation within a defined search space. As such, we do not represent all possible adversarial attacks; if an attack fails then we cannot conclude that the update being verified is valid. The honest population does not have any attack objectives. An honest client with very few examples of the targeted fault can still reduce the probability of that fault using standard optimization techniques applied on his/her own class-mixture. This is why our verification decision is based upon degradation compared to the current model and includes a tolerance mechanism. If we were to require an absolute guarantee that every local model had high-recall on every challenge, then we would confuse specialists with poisoners. Instead, our proposal requires us to ask how a candidate affects the server visible fault responses when assessed at the current training stage.

We separate the attackers' knowledge base into four components:

1.) model knowledge,

2.) challenge family knowledge,

3.) example knowledge,

4.) final test knowledge.

The attacker knows the model and his/her own data. Adaptive attacks also know specific challenge families that have been explicitly created for them. However, they do not know the current rotations samples, nor the final test samples. These constraints are achieved through separate random number generators and data structures. There is no claim made about how long an attacker can remain private regarding a generator given he/she observes client acceptance decisions. Furthermore, we do not attempt to recreate the generator from those decisions in this simulation.

V. DESIGN RATIONALE AND SERVER OPERATION

FIGURE 1 shows where we put our defense in the standard client-server training cycle. FIGURE 2 enlarges the server operation sequence starting with receiving updates up through challenge scoring and weighted combination. We add nothing to the client training loop; we just change how the server assesses each candidate and how much weight the server assigns to each candidate. FIGURE 3 depicts the intended rotation rule. Challenge families are not simply different names for repeated examples; the intervention for challenge family differs at least in terms of sensor response, level of intensity, and regime. Precise detail about the feature-level construction used in our experiment is found in the Implementation Section. At time t, the server begins with one global parameter vector and one trust state for every client. All local candidate models are produced using the same global parameter vectors. The server chooses a fixed or rotating challenge bank based upon the method being tested. The global model is tested on that challenge bank once as a reference. Then, candidate testing measures the degree of degradation relative to that reference. That pairing accounts for differences in inherent difficulty among some challenges as well as changes to the reference model during training. Although the score should not be viewed as a probability that a client is attacking, it represents a deterministic measure of a particular behavioral difference under sample conditions. Normalization of scores allows for a comparable numerical range for weighting; however, it cannot convert empirical calibration into a warranty. A client's locally trained model may preserve behavior observed during challenge testing (i.e., avoid detection) even if it is maliciously behaving. Conversely, an honest client may be penalized due to a legitimate trade-off it made in its local training on the challenge bank. Both scenarios are measured rather than obscured by a binary trust label. Benign stability is a tolerance policy, not a claim that unusual updates are benign. The allowance will prevent small reductions in challenged behavior from leading to the most severe penalties. It is applied uniformly to all clients whether or not the simulator knows that the clients are malicious. Evaluation metrics are derived solely from ground truth attacker labels. Therefore, there is no oracle providing the mechanism with knowledge of which clients need protection. The effect of the mechanism must be determined through both the no-stability ablation and through clean high-heterogeneity experiments. Trust tracks a discounted history of bounded behavioral scores over time. Low scores result in lower influence in subsequent steps; later satisfactory behavior can restore influence. This bounded approach avoids having negative weights and unbounded reputation growth. One potential disadvantage of this approach is delay: an initially trusted client can begin an attack prior to its history decaying; an honest client can remain down-weighted after a brief but temporary mismatch. Therefore, the study reports update decision timing separately from overall global performance at each iteration. There is no single aggregated success metric that can determine if a temporal estimator

This publication is licensed under Creative Commons Attribution CC BY.

responded slowly. Each method uses the same local learning budget. The median coordinate-wise baseline provides an example of a parameter space comparison via implemented parameters; fixed and rotating verification provide a direct comparison of behaviors. Full ChallengeShift-FL incorporates rotation, temporal trust, and stability. Removal of rotation while retaining both temporal trust and stability identifies reuse of the challenge bank. Removal of stability while retaining both rotation and temporal trust identifies tolerance. Rotation of verification without temporal trust isolates instantaneous scores from memory. This structure of comparisons is required since an improvement versus FedAvg alone would not isolate what component of the improvement was responsible for the improvement.

FIGURE 4 explicitly separates between malicious control and benign heterogeneity. FIGURE 5 lists each sequential step including dataset generation, train-test separation, allocation of heterogeneous data across clients, repeated attacks, and final evaluation distribution used for matched methods within each run of the experiment. Randomness is introduced independently for each draw from challenge distributions. Comparison on fair grounds requires that the fixed challenge bank is provided with no additional information compared to the rotating challenge bank. The rotating challenge bank was also not tuned on the final test set. The model selection question and final evaluation question remain distinct within the experimental protocol.

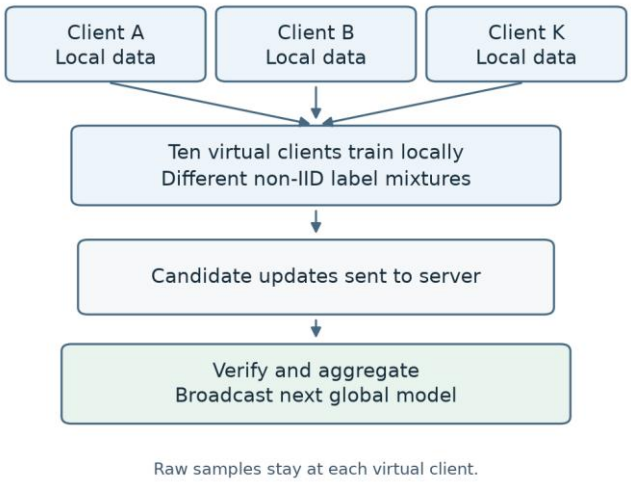

Figure 1. Virtual client-server federation. Raw synthetic windows remain at clients; model candidates are available to the server.

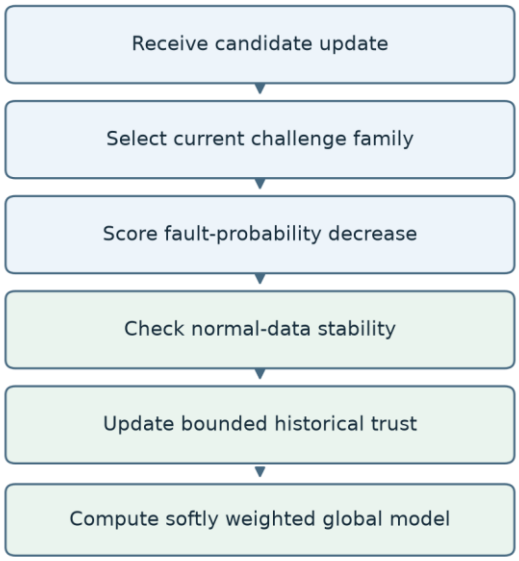

Figure 2. ChallengeShift-FL server sequence. Ground-truth attacker identities are used only for evaluation.

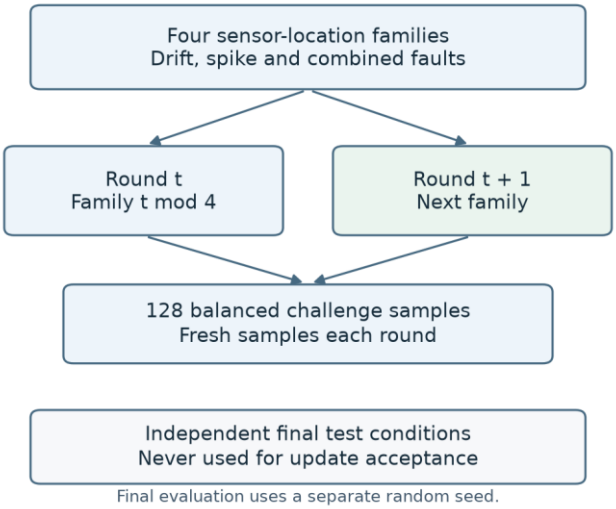

Figure 3. Fixed reuse versus rotating family selection. The experiment cycles four sensor-pair families with fresh draws.

This publication is licensed under Creative Commons Attribution CC BY.

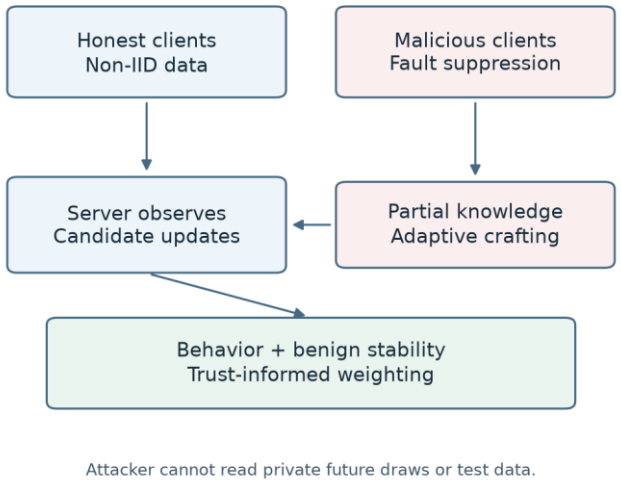

Figure 4. Threat boundary: malicious supervision or parameter control versus legitimate class and regime heterogeneity.

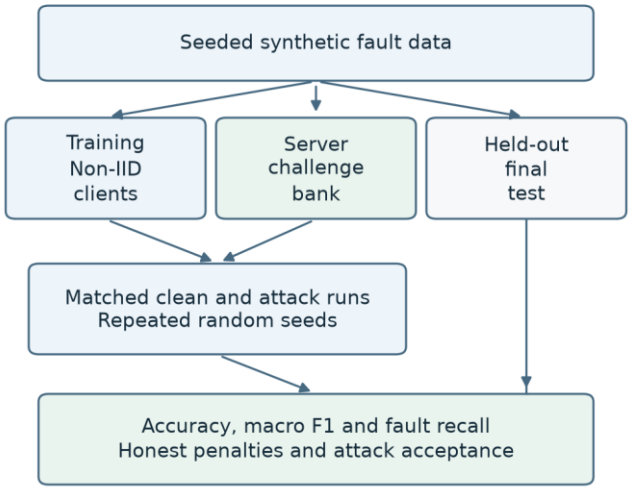

Figure 5. Executed protocol with disjoint generation streams for training, server validation, attacker-known data, and final testing.

VI. IMPLEMENTATION AND EXPERIMENTAL PROTOCOL

a) Executed simulation specification (original study)

This is a proof-of-concept experiment on an explicitly synthetic industrial sensor-window surrogate. It does not use a public industrial dataset, does not constitute a digital twin, and does not establish deployed industrial cybersecurity performance. The source is the supplied simulation code; complete per-seed outputs, convergence trades, platform information and aggregation are in the supplied result record. Reproduce using Python with NumPy and Matplotlib. No downloaded dataset is required. There is no test-driven choice of threshold or selection of the best round.

b) Synthetic data construction and separation

Each of ten clients contains 320 independently generated labeled windows. Each window has eight sensor channels and twelve time steps. Normal windows are Gaussian noise with standard deviation 0.18 and a shared operating-regime offset 0.15r, where r is uniform on [-1,1]. Each client additionally has its own eight-channel offset vector drawn from $N(0,0.07^2)$. The four classes are normal (0), sustained drift (1), short spike (2), and combined drift-plus-spike (3). Target class 1 is the attack objective. For family f in {0,1,2,3}, drift is applied to sensor 2f and spike to sensor 2f+1. The fault duration d is uniform among integers 4,...,12. Drift adds intensity a to the final d samples; a spike adds 2a to the final max(1,floor(d/4)) samples. A combined fault applies both. Training and validation intensity is uniform on [0.8,1.2]. Thus location, intensity, operating regime, duration and combinations all vary explicitly. The semantics are generated from stipulated interventions rather than inferred industrial physics.

Feature extraction concatenates the eight channel means, eight maxima, eight standard deviations, operating regime r, duration d/12, and a constant bias feature. This is 27 features. Duration and regime are supplied as contextual inputs and must be available in any future deployment using this representation. The softmax classifier has four outputs and 108 trainable float64 parameters. It has no hidden layer; this keeps the experiment tractable and substantially limits model expressiveness.

Client class probabilities are independently drawn from a symmetric four-class Dirichlet distribution with concentration 0.8 (moderate) or 0.15 (high heterogeneity). Each client then samples labels from its own probability vector. Client feature offsets create additional covariate differences. Actual sampled priors, average class entropy, and total-variation distance from the uniform class prior are recorded for every seed and condition. All client sample counts are equal, so sample-weighted FedAvg equals the arithmetic average.

The final evaluation contains 2,400 balanced examples per seed, generated independently with a random stream offset by 100,000. It samples all four locations and uses expanded intensity support [0.55,1.55]. This includes both overlapping and previously unseen intensity values; it is NOT an entirely held-out challenge family and NOT exclusively out-of-support data. Training, fixed challenges, normal validation, attacker training, attacker all-family examples and rotating challenges have separate random streams. Fixed validation uses 128 balanced examples from family 0, with stream seed+200,000. Normal validation uses 128 independently generated normal examples, with stream seed+300,000. Adaptive target training uses 320 independently generated balanced examples, with stream seed+400,000. Extensive attacker knowledge adds 512 independently generated all-family examples (seed+500,000) to the entire fixed bank, creating a strict superset of limited knowledge. Per-round rotating challenges contain 128 balanced examples from family r mod 4, with fresh randomness seed+600,000+r. No attacker has the per-round challenge seed or final evaluation data. The four-location schedule is deterministic and may be predictable; this is family rotation with refreshed samples, not a cryptographically secret schedule.

c) Local objective and aggregation

Let W_t be the 27×4 global matrix, x an extracted feature vector, $p_W(k|x)=\text{softmax}(xW)_k$, and D_i the client dataset. The local objective is $L_i(W)=-(1/n_i)\sum_{(x,y \text{ in } D_i)} \log p_W(y|x) +(\lambda/2)\|W_{nonbias}\|^2$, with $\lambda=0.0001$. Each honest client begins from W_t and performs four full-batch gradient steps with

This publication is licensed under Creative Commons Attribution CC BY.

learning rate η=0.18. Thus local steps are four full-batch epochs; there is no mini-batch sampling. Every client participates every round, with synchronous reliable communication, no dropped clients, and no secure aggregation. Verification requires individual candidate models and is incompatible with an unmodified secure-aggregation protocol that hides them. Baseline FedAvg is $W_{t+1}=\sum_i n_i W_{i,t}/\sum_i n_i$. Coordinate median computes the component-wise median of the ten candidates; no guarantee is asserted under this heterogeneous distribution.

d) Threat model and attacks

Clients 0 and 1 are attacker controlled after six initial honest rounds, so the malicious fraction is 20%. Attacks are active at zero-indexed rounds 6,...,23. Attackers see the current global model and control their local data or submitted parameter matrix. They do not alter honest clients, the server, normal validation or final evaluation. There is no Byzantine network attacker, Sybil participation, coordinated multi-client search, gradient inversion, or physical sabotage.

For label flipping, all local class-1 labels on malicious clients are changed to normal; other labels remain intact. The poisoned candidate is trained with the same four local steps. The high non-IID attacked scenario uses this same attack. For replacement, each attacker trains twenty full-batch steps on its flipped local dataset and submits $W_{t+4}(W_{poison}-W_t)$. This is a simple scaled replacement-style poisoning update, rather than a claim of reproducing any published optimizer exactly.

Adaptive attacks begin from the honest local candidate W_h. They optimize $J(W)=CE(W;D_{target\to normal})+3\,KL(p_{W_h}(\cdot|X_{known})\|p_W(\cdot|X_{known}))+0.15\,CE(W;D_i)$, using 25 full-batch gradient steps with step size 0.12. The target attack set comprises class-1 examples from the independently generated attacker dataset. The KL preservation term has the exact softmax cross-entropy gradient $X_{known}^T(p_W-p_{W_h})/n_{known}$. Its teacher is the honest local candidate, rather than an oracle of final-test labels. Both knowledge levels use identical coefficients, step counts and norm budget. The final displacement from W_t is clipped to norm $B=\max(4\|W_h-W_t\|_F,0.35)$. Limited knowledge reveals the entire fixed family-0 bank. Extensive knowledge reveals that bank plus independent labeled examples from all four families, but still not actual rotating samples. Consequently fixed-bank reuse is deliberately a public/reusable validation stress condition; claims must be conditioned on this assumption. This is an implemented differentiable evasion attack, not a full optimizer for every defense's trust trajectory.

e) Behavioral score, stability and trust

For the current challenge bank C_t and each fault class k in {1,2,3}, let $C_{t,k}$ be its 32 examples. The degradation score is $d_{i,t}=\max_k\,[\,(1/|C_{t,k}|)\sum_{x\in C_{t,k}}(p_{W_t}(k|x)-p_{W_{i,t}}(k|x))\,]$. Scores measure expected correct-class probability loss relative to the current global model, and lie within [-1,1]. Negative values reflect improvement. This is a relative behavioral measure; it does not certify that the global model is already correct or detect every gradual accumulated suppression.

Let m_t be the median of ten degradation scores, and MAD_t the median absolute deviation from m_t. The base tolerance is $\theta_t=\max(0.025,m_t+2.5\times1.4826\times MAD_t)$. The robust scale normalizer is heuristic; no normality assumption or theoretical false-positive rate follows. The tolerance is estimated from all candidates and can be contaminated by malicious candidates. It is not an oracle based on attacker identities. For the benign-stability mechanism, calculate $e_{i,t}=CE(W_{i,t};V_0)-CE(W_t;V_0)$ on the 128 normal-only validation examples. A candidate is stable if $e_{i,t}\le0.02$. Its allowed degradation becomes $a_{i,t}=2\theta_t$; otherwise $a_{i,t}=\theta_t$. The no-stability variants always use θ_t. This explicit allowance is meant to tolerate heterogeneity, but normal-only stability can also benefit a fault-suppression attacker. The experiment must assess that tradeoff rather than presuppose protection.

Define quality $q_{i,t}=\exp[-\max(d_{i,t}-a_{i,t},0)/0.04]$, bounded in (0,1]. An instantaneous suspicious flag is $q_{i,t}<0.5$. Fixed and Rotating use weights $\max(q_{i,t},0.05)$. Temporal variants initialize $\tau_{i,0}=1$, then update $\tau_{i,t}=0.8\tau_{i,t-1}+0.2q_{i,t}$, and use weights $\max(\tau_{i,t},0.05)$. Trust is a bounded smoothing heuristic, not a calibrated posterior probability of honesty. Its inertia can delay response to newly malicious clients. The weighted global model is $W_{t+1}=\sum_i w_{i,t}W_{i,t}/\sum_i w_{i,t}$. No candidate is excluded entirely. Honest false rejection is therefore zero by construction; honest false flags and influence reduction are the meaningful fairness diagnostics here. A zero rejection rate cannot by itself establish fairness.

f) Metrics and cost accounting

The final multiclass confusion matrix gives accuracy, per-class recall, precision and macro F1 (the arithmetic mean of four class F1 scores, with zero-denominator protection). Mean fault-class recall averages recalls for classes 1,2,3. Missed-fault rate is the fraction of all fault-class final-test examples predicted normal. Target ASR is # {y=1,prediction=0}/# {y=1}; it is the target-to-normal error rate, reported even in clean controls, not a conditional increase over clean correctness. Honest false flag rate is the number of honest client-rounds with $q<0.5$ divided by all honest client-rounds. Malicious unflagged rate is the fraction of malicious client-rounds with $q\ge0.5$. Baselines without a detector mark no client suspicious, so their unflagged rate is one by convention; this does not measure coordinate median's aggregation resilience. There are 36 malicious client-rounds and 204 honest client-rounds in attacked runs; all-honest runs have 240 honest client-rounds.

Final model selection always uses round 24. Accuracy is also calculated after each round to document convergence; these diagnostics do not alter training, hyperparameters or stopping. Communication assumes dense float64 download of the 108-parameter global model and upload of each candidate each round: $2\times10\times24\times108\times8=414{,}720$ bytes per run. Model payload communication is identical for all methods; metadata, transport, initialization and attack-side knowledge transfer are not counted. No extra client message is required for server-generated challenges. Server verification/aggregation timing is the measured elapsed time after local candidates are obtained, including challenge generation, scoring, trust and aggregation. Baseline times include their aggregation alone, so differences estimate additional server work but are not hardware-

This publication is licensed under Creative Commons Attribution CC BY.

independent performance measures. Local/attack computation is timed separately. Each candidate challenge prediction costs O(|C_t|×27×4), giving server work O(K|C_t|PC) plus optional normal validation O(K|V_0|PC) for K clients, P input features and C classes. Matrix storage is O(KPC); this prototype has no measured deployment memory benchmark.

Table I. Executed method components. All weights remain positive.

Method	Rotation	Memory / stability
FedAvg	No	Neither
Median	No	Coordinate median
Fixed	No	Immediate / no
Rotating	Yes	Immediate / no
ChallengeShift	Yes	Trust / yes
NoRotation	No	Trust / yes
NoStability	Yes	Trust / no

Table II. Main configuration. All values correspond to executed code.

Setting	Value
Clients / attackers	10 / 2
Rounds / local steps	24 / 4 full-batch
Model	27 × 4 softmax; 108 parameters
Data / final test	320 per client / 2,400
Seeds	11, 29, 47
Dirichlet α	.8 moderate; .15 high
Challenges / normal bank	128 / 128 windows
Step / trust decay	.18 / .8
Tolerance / scale	.025 / .04
Trust floor / attack start	.05 / round index 6

VII. RESULTS AND ABLATION ANALYSIS

1. *Results: executed evidence*

Completed all 147 scheduled runs using seven different methods, seven scenarios and three independently seeded runs (seeds 11, 29 and 47) with each run completing twenty-four synchronous rounds resulting in a total of 3528 completed rounds. All processes including data generation, client local learning, attacker optimization, server weighting and final analysis were implemented in Python rather than based upon an abstract conceptual model. The experimental loop time was approximately seventy point two-seven seconds as measured against the available execution environment, not including subsequent plots. The operating system was Windows 11 with the version of python being 3.12.14, NumPy 2.3.5 and AMD64 Family 25 Model 80 Stepping 0, AuthenticAMD; no processor product name was identified by the processor string nor was there inference to the number of cores or hardware utilization.

Numerical results listed hereunder represent simple arithmetic averages and sample standard deviation of the three separately created client population datasets. No p-value or claim to statistical significance has been made.

No advantage has been found for rotating validation in comparison to fixed validation in these experiments. The primary result supporting this conclusion is a larger and more pronounced effect size for the full proposed mechanism's inclusion of both normal and temporal smoothing which preserves poisoning effects under most conditions. Figure 6 shows the target-fault-to-normal error as it relates to attacks while figure 7 illustrates a direct comparison of the honest-only control group for predictive accuracy and suspicious-update flag identification. Error bars represent true variability across multiple seeds, not due to repeated measurement of one seed. Due to significant variance in many cases, individual mean differences may be weak at best.

2. *Clean learning and the heterogeneity control*

With honest participation and moderate non-IID data, FedAvg reached 74.00 ± 10.84% accuracy and approximately 0.700 mean macro F1. Its mean fault-class recall was 83.26 ± 15.99%, and 0.81 ± 0.67% of all fault examples were classified as normal. Target class-1 error specifically was 1.06 ± 1.42%. Complete ChallengeShift reached 73.99 ± 10.83% accuracy, approximately 0.700 macro F1, 83.24 ± 15.98% mean fault recall and the same 0.81 ± 0.67% missed-fault rate. These near-identical averages show that the complete mechanism changed little in this clean setting; they do not establish equivalent behavior under all distributions.

Coordinate median obtained approximately 0.595 macro F1, below FedAvg's clean value. Fixed behavioral weighting obtained approximately 0.668, rotating weighting 0.688, and rotating trust without stability 0.683. The class heterogeneity therefore mattered even before an attacker was introduced. Averaged across seed populations, the mean client class entropy fell from 0.999 natural-log units with Dirichlet concentration 0.8 to 0.441 with concentration 0.15. The mean total-variation distance between a client's class probabilities and the uniform prior increased from 0.361 to 0.598. These are calculated characteristics of the sampled client priors, not estimates from a real industrial deployment.

Under honest-only high non-IID conditions, FedAvg macro F1 fell to approximately 0.605 and coordinate median to 0.437. Complete ChallengeShift achieved approximately 0.604; the no-rotation counterpart achieved 0.605. Mean target error rose to 18.61 ± 30.66% for FedAvg and 19.00 ± 31.33% for ChallengeShift even without poisoning. Thus high target error in an attacked condition cannot be attributed entirely to the attacker. The honest-only high-heterogeneity control supplies an essential reference for separating distribution difficulty from adversarial degradation.

3. *Basic targeted poisoning and replacement*

Targeted label flipping increased FedAvg target error to 36.78 ± 38.06%, compared with 1.06 ± 1.42% in its clean moderate control. Coordinate median achieved 17.78 ± 23.76%. Fixed behavioral weighting achieved 35.17 ± 40.06%, whereas rotating weighting, complete ChallengeShift, NoRotation and NoStability each achieved 36.78 ± 38.06%. Rotation therefore provided no favorable mean result here. The instantaneous rotating-minus-

This publication is licensed under Creative Commons Attribution CC BY.

fixed contrast was +1.61 ± 2.79 percentage points. With trust and stability both present, the matched rotation contrast was exactly zero across all three seeds. The extreme dispersion cautions against claiming reliable protection from either a modest average advantage or one favorable seed.

Scaled replacement was much more damaging to FedAvg: target error was 99.89 ± 0.19%, with approximately 0.478 macro F1. Coordinate median attained 21.22 ± 29.13% target error and approximately 0.652 macro F1. Fixed behavioral weighting reached 39.28 ± 50.62% target error and approximately 0.706 macro F1; rotating weighting reached 40.11 ± 52.41% and approximately 0.689. Their paired rotation contrast was +0.83 ± 1.89 percentage points. The fixed-versus-rotating means differed little relative to the dispersion, and rotation did not improve the mean target outcome.

The full ChallengeShift and its no-rotation ablation both reached 99.72 ± 0.25% target error under replacement, with approximately 0.480 macro F1. Removing the stability allowance lowered target error to 77.94 ± 28.14% and increased macro F1 to approximately 0.620, but still left substantial suppression. Removing temporal trust as well, by comparing NoStability with the instantaneous Rotating method, lowered mean target error further to 40.11%. These executed ablations indicate that permissive normal-only stability and delayed temporal response can undermine this particular defense construction. They do not demonstrate that all stability mechanisms or all trust estimators are harmful.

4. *Adaptive knowledge and the primary rotation question*

When the attackers had minimal knowledge of the model, the FedAvg target error was 67.33 ± 17.11%, the median of the coordinates was 19.61 ± 26.02%, the fixed weightings were 54.22 ± 39.69%, and the rotating weightings were 53.78 ± 39.91%. The paired difference between the rotating and fixed weights was –0.44 ± 0.38 percentage points. There was a relatively small (only 0.67 percentage point) improvement in two seeds through rotation, and no change to the third. That such a small gain occurred for this particular problem does not demonstrate that there is an advantage due to rotation under a broad range of problems. A full implementation of ChallengeShift resulted in an average error rate of 66.56 ± 18.44%, which is significantly better than the NoRotation approach at an average error rate of 60.72 ± 28.47%. As such, the rotation-based method used within ChallengeShift showed a significant rotation-based contrast of +5.83 ± 10.10 percentage points when compared against the NoRotation approach; although the impact was large, it came from just one seed. The NoStability approach resulted in an average error rate of 55.44 ± 37.17%.

In addition to having the extensive attacker knowledge that consisted of all of the examples from the limited-knowledge bank, we also created additional examples based on each family independently (for example, if we were using the Family A dataset then we would create another set of examples). In the case where we have an attacker with extensive knowledge about our model, we see that the FedAvg target error is 64.67 ± 35.15%, the median is 23.28 ± 31.89%, and both the fixed and rotating weightings are 55.50 ± 49.87% and 55.56 ± 49.78% respectively. When looking at the rotation-based contrast, we find that it is +0.06 ± 0.10 percentage points, which indicates little to no benefit gained from rotation when comparing these two approaches. We can also compare the performance of NoRotation versus ChallengeShift at achieving lower attack success rates, and we see that while ChallengeShift has a slightly higher average error rate (61.83 ± 39.60%) than NoRotation (59.33 ± 43.61%), it provides a larger contrast (+2.50 ± 4.33 percentage points), indicating that ChallengeShift provides a greater defense against attacks.

While it might seem like increasing the amount of knowledge available to the attacker would monotonically improve their ability to carry out successful attacks, this experiment shows otherwise. While it is possible to explain why this may or may not happen (such as due to finite-step optimizers, preservation objectives and trajectory dependent bounds on norms), this experiment does not isolate these explanations. Therefore, as we continue to study how to measure how well an optimization process defends against adversarial attacks, we will need to examine the strength of how effective different forms of optimizations can be, and until we do so, we cannot conclude that increases in knowledge always lead to increased effectiveness in carrying out attacks.

5. *Fairness diagnostics and computational cost*

In the all-honest moderate control, 2.64% of honest client-rounds were identified as dishonest (or suspect) by the fixed weighting strategy, 1.11% by the rotating weighting approach, and 1.67% by NoStability; meanwhile, the ChallengeShift and NoRotation strategies did not identify any. In the all-honest high non-IID control, the percentage of client-rounds suspected to be dishonest were 2.50%, 1.39%, 1.53%, 0.69%, and 0% respectively. While reducing suspicions, this reduction in thresholds allows for potentially destructive replacement updates. When using replacement, the malicious client that went undetected was at a 23.15% rate for fixed weighting, a 27.78% rate for rotating weighting, a 87.04% rate for full ChallengeShift, and a 38.89% rate for NoStability. As such, these diagnostic rates need to be evaluated in conjunction with the ultimate predictions generated from each of the models; they are not indicative of a model's security status.

The aggregated weights will always remain positive, and the least value that can be assigned to a weight is 0.05. Therefore, there are no submitted models that are completely eliminated. Therefore, actual false positives (in the form of rejecting honest clients) is zero, while actual true negatives (i.e., including malicious clients in the training set) is 100% for all methods. A legitimate (and meaningful) measure of fairness will be an honest suspicious-flag rate, whereas a valid measure of detection will be a malicious unflagged rate. Both measures avoid misinterpreting the nature of soft weighting as either an absolute acceptance or rejection mechanism. The median will have no flagging on the coordinates; therefore, it will have a 100% unflagged rate, even though it still provides some level of robustness when considering the aggregate data from many scenarios. Each method had equal amounts of payload communicated during model uploads/downloads (under the given dense float64 convention), i.e., 414,720 bytes per run. In the clean moderate control environment, the time spent by the server for both verifying and aggregating results after 24 rounds of

This publication is licensed under Creative Commons Attribution CC BY.

communication was roughly .00089 seconds for FedAvg, .00428 seconds for median, .04775 seconds for fixed weighting, .12144 seconds for rotating weighting, .12788 seconds for full ChallengeShift, .08947 seconds for NoRotation and .12911 seconds for NoStability. These times include Challenge generation performed on the server side. These values represent a very small local simulation and should not be used to determine any type of operational latency guarantees.

6. *Interpretive limitations and next experiments*

Current evidence refutes the claim that applying rotation to the randomization process will enhance protection for an application against poisoning attacks (and also preserve fairness). Current evidence does support the following more narrow interpretation: This specific implementation is showing some potential trade-offs and failure modes that would need stronger verification protocols prior to deployment. Fixed bank used one family of seed values and used same seed value every time it was called; Rotating bank utilized four different families of seed values and new seed values each time it was called. Rotation is therefore both confusingly related to "family" coverage and "sample refresh." Future study using a full-factorial experimental design could include comparing fixed multi-family banks, a fixed bank generating random samples from a single fixed family, and a rotating bank with matching exposure and validation costs. In addition to the above priorities there are many additional issues such as having real world data provided by the public sector, richer temporal models, multiple seeds, control differences in terms of ASR that can be differentiated from normal data, independent calibration of tolerance levels, and adaptive optimization over the entire trust trajectory. None of the preliminary findings here are intended to provide any type of novelty priority or any defense guarantees.

MATHEMATICAL SPECIFICATION

$$p_W(k|x) = softmax(xW)_k$$

$$L_i(W) = -n_i^{-1} \Sigma \log p_W(y|x) + \lambda||W||^2/2$$

$$W_{t+1} = \Sigma_i n_i W_{i,t} / \Sigma_i n_i$$

$$x = \varphi(S + 0.15r + \delta_{f,a,d,y})$$

$$d_i = max_k \, mean_{C_k}[p_t(k|x) - p_i(k|x)]$$

$$\theta = max(0.025, median(d) + 3.7065 \, MAD(d))$$

$$a_i = 2\theta \; if \; normal \; CE \; change \leq 0.02; \; else \; \theta$$

$$q_i = exp[-max(d_i - a_i, 0)/0.04]$$

$$\tau_{i,t} = 0.8\tau_{i,t-1} + 0.2q_{i,t}$$

$$w_i = max(\tau_i, 0.05); \; W_{next} = \Sigma w_i \, W_i / \Sigma w_i$$

$$ASR = count(y = 1, prediction = 0)/count(y = 1)$$

$$HFR = honest \; flags \; / \; honest \; client - rounds$$

$$HRR = excluded \; honest \; updates \; / \; honest \; updates = 0$$

The parameters of the model contain four parts. First part contains φ which is used to obtain the mean, maximum, standard deviation, and context. Second parameter is S for the Gaussian sensor window. Third parameter is r for operation mode. Fourth parameter is δ for the initial synthetic fault intervention with index of family f, intensity a, duration d, and label y. Fifth parameter is λ for exclusion of the bias row. Sixth parameter is MAD (median absolute deviation). Seventh parameter is τ for heuristics that bound the trust. Eighth parameter is HFR (false flag rate) and ninth parameter is HRR (hard reject rate). All of the other parameters were given previously in Section VI.

Table III. Target fault-to-normal rate (%), mean ± sample SD across three seeds. Lower is better.

Method	Label flip	Replacement	Adaptive limited	Adaptive extensive	High non-IID
FedAvg	36.78 ± 38.06	99.89 ± 0.19	67.33 ± 17.11	64.67 ± 35.15	78.11 ± 22.99
Median	17.78 ± 23.76	21.22 ± 29.13	19.61 ± 26.02	23.28 ± 31.89	72.39 ± 47.54
Fixed	35.17 ± 40.06	39.28 ± 50.62	54.22 ± 39.69	55.50 ± 49.87	78.11 ± 22.99
Rotating	36.78 ± 38.06	40.11 ± 52.41	53.78 ± 39.91	55.56 ± 49.78	78.39 ± 22.46
ChallengeShift	36.78 ± 38.06	99.72 ± 0.25	66.56 ± 18.44	61.83 ± 39.60	78.78 ± 21.95
NoRotation	36.78 ± 38.06	99.72 ± 0.25	60.72 ± 28.47	59.33 ± 43.61	78.11 ± 22.99
NoStability	36.78 ± 38.06	77.94 ± 28.14	55.44 ± 37.17	55.67 ± 49.59	78.78 ± 21.95

Table IV. Adaptive-limited model performance and update diagnostics. Percentages except macro F1.

Method	Accuracy	Macro F1	Fault recall	Missed faults	Honest flags	Malicious unflagged
FedAvg	64.40 ± 9.55	0.605 ± 0.126	60.41 ± 9.70	30.19 ± 0.69	0.00 ± 0.00	100.00 ± 0.00
Median	69.49 ± 4.46	0.634 ± 0.085	67.67 ± 15.22	10.96 ± 8.96	0.00 ± 0.00	100.00 ± 0.00
Fixed	64.15 ± 9.98	0.594 ± 0.145	64.89 ± 1.98	22.78 ± 13.52	1.96 ± 3.40	83.33 ± 28.87
Rotating	65.92 ± 8.67	0.613 ± 0.127	65.22 ± 1.91	22.93 ± 12.98	0.33 ± 0.57	77.78 ± 27.36
ChallengeShift	64.47 ± 9.43	0.605 ± 0.125	60.72 ± 9.15	29.74 ± 1.46	0.00 ± 0.00	98.15 ± 3.21
NoRotation	64.46 ± 9.46	0.604 ± 0.127	62.69 ± 5.76	26.41 ± 7.23	0.00 ± 0.00	92.59 ± 12.83
NoStability	64.94 ± 9.48	0.605 ± 0.134	64.63 ± 2.94	23.57 ± 11.76	0.16 ± 0.28	81.48 ± 27.40

Table V. All-honest controls: macro F1 and honest false flags (%). No client is excluded.

Method	Moderate F1	High F1	Moderate flags	High flags
FedAvg	0.700 ± 0.154	0.605 ± 0.027	0.00 ± 0.00	0.00 ± 0.00

This publication is licensed under Creative Commons Attribution CC BY.

Method	Moderate F1	High F1	Moderate flags	High flags
Median	0.595 ± 0.099	0.437 ± 0.111	0.00 ± 0.00	0.00 ± 0.00
Fixed	0.668 ± 0.209	0.539 ± 0.093	2.64 ± 4.57	2.50 ± 4.33
Rotating	0.688 ± 0.174	0.589 ± 0.018	1.11 ± 1.92	1.39 ± 1.46
ChallengeShift	0.700 ± 0.154	0.604 ± 0.028	0.00 ± 0.00	0.69 ± 1.20
NoRotation	0.700 ± 0.154	0.605 ± 0.027	0.00 ± 0.00	0.00 ± 0.00
NoStability	0.683 ± 0.183	0.596 ± 0.019	1.67 ± 2.89	1.53 ± 1.34

Table VI. Server aggregation/verification time per 24-round adaptive-limited run (seconds).

Method	Time mean ± SD
FedAvg	0.002 ± 0.001
Median	0.005 ± 0.001
Fixed	0.074 ± 0.005
Rotating	0.119 ± 0.013
ChallengeShift	0.144 ± 0.011
NoRotation	0.094 ± 0.009
NoStability	0.123 ± 0.005

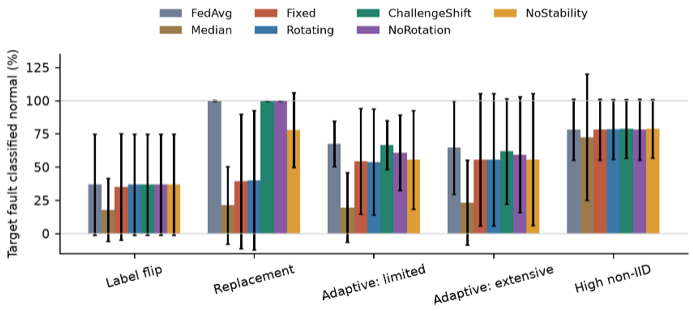

Figure 6. Measured target fault-to-normal rates. Bars show three-seed means; whiskers show sample SD, not confidence bounds. SD intervals can extend beyond the valid 0–100% outcome range.

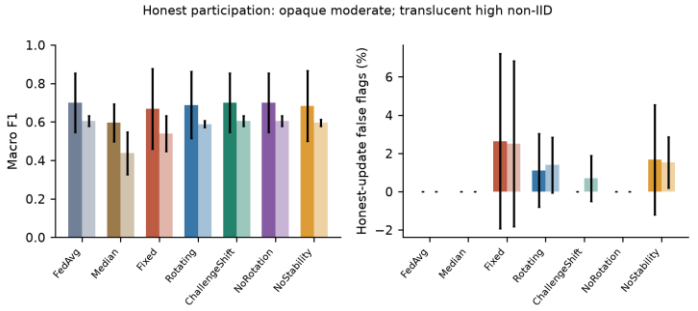

Figure 7. All-honest moderate and severe heterogeneity controls. Honest flags measure downweighting, not exclusion; error bars summarize three seeds.

Table I lists the implemented modules; Table II describes the training environment. Tables III-V outline measured results of attacks on our system (with adaptive limitations), as well as control measurements from an "honest-only" mode. Table VI reports the average processing time per request for the servers.

VIII. INTERPRETATION AND VALIDITY

The central result must be read as a comparison within a small synthetic model class. When fixed and rotating verification differ, the difference can arise from coverage of the fault directions, the difficulty of the sampled challenge conditions, or interactions between weighting and non-IID optimization. Matched seeds reduce random initialization differences but do not remove these explanations. The no-rotation ablation is the most direct estimate of rotation's contribution within the complete system. The fixed-versus-rotating immediate-score comparison provides a second check without trust memory. A defense can reduce malicious acceptance without reducing final attack success. The retained malicious contributions may still have large influence, or the honest population may have too little data to recover the targeted class. Conversely, a model can have acceptable target recall even when some malicious updates pass because the honest updates counteract them. For this reason, the manuscript reports malicious acceptance as an update-level diagnostic and targeted attack success as a model-level outcome. They answer different questions and should not be substituted for one another.

Honest-client fairness is measured here as treatment of honest contributions, not as a complete distributive fairness analysis. A low rate of honest penalties does not establish equal performance across factories, devices, operating regimes, or demographic groups. The stronger test would report each client's held-out fault recall and compare losses across different participation schedules. The present all-client schedule and common final test set simplify causal interpretation of the aggregation comparison, but they cannot resolve the experience of an infrequently sampled site or a rare regime absent from server challenges.

The classifier is a transparent low-capacity surrogate. Its linear decision boundaries make optimization reproducible and allow the attacker preservation objective to be inspected, but they constrain both defense and attack. A deep temporal model could learn a different representation and a more flexible backdoor. An adversary could preserve all challenge directions yet alter an unmeasured feature interaction. The observed experiment

This publication is licensed under Creative Commons Attribution CC BY.

therefore tests the stated mechanism and search routine, not the full adversarial space of a production federated fault-diagnosis service.

The generated sensor-window interventions specify class labels by construction. That is legitimate for a synthetic benchmark, but it does not establish the physical validity of the label under a real process. Duration and operating-regime features are surrogates, not measurements from a digital twin calibrated to a plant. The generator contains no conservation equations, process control loop, or identified plant dynamics. A physically validated challenge generator would need to document its operating envelope and demonstrate that fault labels remain correct over the sampled intensity range before challenge scores could support industrial decisions.

Timing results measure executed software on the available environment. They include the implemented server evaluations and should not be interpreted as a hardware-independent throughput guarantee. Challenge scoring cost depends on the number of clients, samples, and model parameters. The experiment exchanges the same parameter payload in the behavioral methods as in FedAvg because challenges are generated and evaluated at the server. That payload accounting omits network protocol headers, encryption, retries, and validation-set distribution. It therefore describes simulated model communication, not total network traffic.

IX. LIMITATIONS AND FUTURE WORK

The first limitation is external validity. The experiment used synthetic sensor window data rather than independently obtained public industrial fault data. The faults in this experimental setup were designed to have easily recognizable signal characteristics. In addition, there is some degree of class imbalance as well as regime changes. However, we make no claims about how common these types of faults are nor how the covariances of the faults will compare with a real-world system. Thus, our next step should include reproducing the protocol using an available industrial process benchmark (with explicitly defined origin) and separating the training set and the final evaluation set. Regardless of whether the final implementation includes public benchmark testing, prior to its operationally being deployed into production at a specific site, additional validation will be necessary. The second limitation is that the adaptive algorithm has limited ability to detect certain "adversarial" scenarios. The algorithm provides a preservation penalty for those challenge patterns that it knows exist within a given fixed optimization budget. Other possible forms of adversarial behavior include attacks based upon weighting, exploiting trust recovery, coordinating multiple malicious updates, inferring family membership of challenges from acceptance feedback, and attacking rarely occurring states not included in the bank. An interesting direction for further research would involve allocating equal budgets to all methods of searching for attacks and comparing the effectiveness of each method against the defenses rather than having them compete using only one common form of attack. Moreover, if such comparative tuning occurs, then it must be done separately from the final reporting of test results to ensure that a particular type of exploit was tested on only its original development examples.

The third limitation involves the level of detail provided in terms of statistical resolution. Although several independent random seeds allow us to describe mean values and standard deviation values, they do not enable us to determine a consistent ordering among datasets and regimes. Using the same seed allows comparison across experiments, therefore a follow-up analysis would retain paired differences and increase the number of repeated trials. Prior to analyzing the results, it is recommended that both the primary measure (allowed honest-penalty budget), and both the attacker's search power be determined. Otherwise, the most advantageous type of attack or the lowest acceptable risk value may become the focus after the fact.

The fourth limitation is related to calibration. Both the choice of behavioral allowance and the rate at which trust decays are parameters that impact cost depending on the distribution of legitimate updates. Increasing tolerance protects honest experts and permits weak adversaries. Decreasing tolerance improves filtering while increasing concentration of influence among client systems similar to those that comprise the server bank. The present study identifies this trade-off using ablation studies however we did not address determining optimal thresholds for an arbitrary deployment. Calibration on previously unseen benign development clients followed by evaluating prospects with constant thresholds will provide a much stronger basis for prospective evaluations.

Additional research areas that we suggest testing include partial client participation, failure in inter-client communications, and protection of client update privacy. Currently per-client behavioral validation requires candidate identifiers and persistent trust states. If a deployment utilizes secure aggregated updates, the defense would require a corresponding validation protocol rather than merely continuing with its existing scoring function. All challenge schedules should be stored for auditing purposes while access to unreported challenges continues to be limited. Auditing and secrecy are two distinct goals; a good protocol can release generator code and previous schedule information while maintaining new draw requests for future assessments.

X. CONCLUSION

ChallengeShift-FL is an implemented proposal for rotating fault-oriented behavioral checks with benign-stability tolerance and bounded trust-weighted aggregation. The study asks whether rotation adds value over fixed validation and reports that comparison directly instead of inferring it from a contrast with undefended averaging. The evidence consists of executed virtual-client experiments on explicitly synthetic sensor-window data. The results support only the measured conditions and must be interpreted alongside honest-client penalties, attacker knowledge, and the ablation outcomes. The design does not guarantee protection from arbitrary poisoning or establish real-world industrial cybersecurity effectiveness.

This publication is licensed under Creative Commons Attribution CC BY.

REFERENCES

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Aguera y Arcas. "Communication-Efficient Learning of Deep Networks from Decentralized Data." Proceedings of AISTATS, PMLR 54, pp. 1273-1282, 2017. https://proceedings.mlr.press/v54/mcmahan17a.html

[2] Dong Yin, Yudong Chen, Ramchandran Kannan, Peter Bartlett. "Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates." Proceedings of ICML, PMLR 80, pp. 5650-5659, 2018. https://proceedings.mlr.press/v80/yin18a.html

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, Vitaly Shmatikov. "How To Backdoor Federated Learning." Proceedings of AISTATS, PMLR 108, pp. 2938-2948, 2020. https://proceedings.mlr.press/v108/bagdasaryan20a.html

[4] Peter Kairouz, H. Brendan McMahan. "Advances and Open Problems in Federated Learning." Foundations and Trends® in Machine Learning, vol. 14, no. 1-2, 1-210, 2021. doi: 10.1561/2200000083.

[5] Xiaoyu Cao, Minghong Fang, Jia Liu, Neil Zhenqiang Gong. "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping." Proceedings 2021 Network and Distributed System Security Symposium, 2021. doi: 10.14722/ndss.2021.24434.

[6] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, Thomas Schneider. "FLAME: Taming Backdoors in Federated Learning." 31st USENIX Security Symposium, pp. 1415-1432, 2022. https://www.usenix.org/system/files/sec22-nguyen.pdf

[7] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, Ahmad-Reza Sadeghi. "DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection." arXiv preprint arXiv:2201.00763, 2022. doi: 10.48550/arXiv.2201.00763.

[8] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, Neil Zhenqiang Gong. "FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients." arXiv preprint arXiv:2207.09209, 2022. doi: 10.48550/arXiv.2207.09209.

[9] Xingyu Li, Zhe Qu, Shangqing Zhao, Bo Tang, Zhuo Lu, Yao Liu. "LoMar: A Local Defense Against Poisoning Attack on Federated Learning." arXiv preprint arXiv:2201.02873, 2022. doi: 10.48550/arXiv.2201.02873.

[10] Sebastien Andreina, Giorgia Azzurra Marson, Helen Mollering, Ghassan Karame. "BaFFLe: Backdoor Detection via Feedback-based Federated Learning." 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS), 852-863, 2021. doi: 10.1109/icdcs51616.2021.00086.

[11] Chaoyi Zhu, Stefanie Roos, Lydia Y. Chen. "LeadFL: Client Self-Defense against Model Poisoning in Federated Learning." Proceedings of ICML, PMLR 202, pp. 43158-43180, 2023. https://proceedings.mlr.press/v202/zhu23j.html

[12] Yongquan Liang, Suyi Wang, Yuncan Tang. "FedRISC-IIoT: A federated learning poisoning defense framework combining reputation-driven client selection and layered robust aggregation for industrial IoT." Future Generation Computer Systems, vol. 185, 108640, 2026. doi: 10.1016/j.future.2026.108640.

[13] Anji Ayman Hamwi, Meenakshi Mittal. "FL-TWIN: a unified federated learning system for intrusion detection with digital twins modelling." Scientific Reports, vol. 16, no. 1, 28249, 2026. doi: 10.1038/s41598-026-58750-1.

[14] Ammar Alazab, Abin Kumbalapalliyil Tom, Tony Jan, Md Whaiduzzaman, Thien Nguyen, Ansam Khraisat, et al.. "DBST-FL: Dynamic Behavioural and Semantic Trust for Robust Federated Learning in Industrial IoT." Sensors, vol. 26, no. 15, 4712, 2026. doi: 10.3390/s26154712.

[15] Wei Zhang, Xiang Li, Hui Ma, Zhong Luo, Xu Li. "Federated learning for machinery fault diagnosis with dynamic validation and self-supervision." Knowledge-Based Systems, vol. 213, 106679, 2021. doi: 10.1016/j.knosys.2020.106679.

[16] Quan Qian, Bin Zhang, Chuan Li, Yongfang Mao, Yi Qin. "Federated transfer learning for machinery fault diagnosis: A comprehensive review of technique and application." Mechanical Systems and Signal Processing, vol. 223, 111837, 2025. doi: 10.1016/j.ymssp.2024.111837.

[17] Hangyu Zhu, Jinjin Xu, Shiqing Liu, Yaochu Jin. "Federated learning on non-IID data: A survey." Neurocomputing, vol. 465, 371-390, 2021. doi: 10.1016/j.neucom.2021.07.098.

[18] Yichen Wan, Youyang Qu, Wei Ni, Yong Xiang, Longxiang Gao, Ekram Hossain. "Data and Model Poisoning Backdoor Attacks on Wireless Federated Learning, and the Defense Mechanisms: A Comprehensive Survey." IEEE Communications Surveys & Tutorials, vol. 26, no. 3, 1861-1897, 2024. doi: 10.1109/comst.2024.3361451.

[19] Zhiyong Jin, Runhua Xu, Chao Li, Yizhong Liu, Jianxin Li. "Sparsification Under Siege: Defending Against Poisoning Attacks in Communication-Efficient Federated Learning." arXiv preprint arXiv:2505.01454, 2025. doi: 10.48550/arXiv.2505.01454.

[20] Runhua Xu, Shiqi Gao, Chao Li, James Joshi, Jianxin Li. "Dual Defense: Enhancing Privacy and Mitigating Poisoning Attacks in Federated Learning." arXiv preprint arXiv:2502.05547, 2025. doi: 10.48550/arXiv.2502.05547.

[21] Yujing Wang, Hainan Zhang, Sijia Wen, Wangjie Qiu, Binghui Guo. "Defending Against Sophisticated Poisoning Attacks with RL-based Aggregation in Federated Learning." arXiv preprint arXiv:2406.14217, 2024. doi: 10.48550/arXiv.2406.14217.

[22] Heyi Zhang, Yule Liu, Xinlei He, Jun Wu, Tianshuo Cong, Xinyi Huang. "SoK: Benchmarking Poisoning Attacks and Defenses in Federated Learning." arXiv preprint arXiv:2502.03801, 2025. doi: 10.48550/arXiv.2502.03801.

[23] Oussama Akram Bensiah, Rohallah Benaboud. "FedDPA: Dynamic Prototypical Alignment for Federated Learning with Non-IID Data." Electronics, vol. 14, no. 16, 3286, 2025. doi: 10.3390/electronics14163286.

[24] Yang Ge, Yong Ren. "Federated Transfer Fault Diagnosis Method Based on Variational Auto-Encoding with Few-Shot Learning." Mathematics, vol. 12, no. 13, 2142, 2024. doi: 10.3390/math12132142.

[25] Funa Zhou, Yi Yang, Chaoge Wang, Xiong Hu. "Federated Learning Based Fault Diagnosis Driven by Intra-Client Imbalance Degree." Entropy, vol. 25, no. 4, 606, 2023. doi: 10.3390/e25040606.

[26] Yuting Xu, Wangzhuo Yang, Shuwang Du, Meifu Zhang. "Federated Learning Based on Fuzzy Fusion Rules for Chemical Production Process Fault Diagnosis." Sensors, vol. 26, no. 11, 3545, 2026. doi: 10.3390/s26113545.

[27] Wanqian Yang, Gang Yu. "Federated Multi-Model Transfer Learning-Based Fault Diagnosis with Peer-to-Peer Network for Wind Turbine Cluster." Machines, vol. 10, no. 11, 972, 2022. doi: 10.3390/machines10110972.

[28] Yingying Liang, Peng Zhao, Yimeng Wang. "Federated Few-Shot Learning-Based Machinery Fault Diagnosis in the Industrial Internet of Things." Applied Sciences, vol. 13, no. 18, 10458, 2023. doi: 10.3390/app131810458.

[29] Yanxin Hu, Xiaoman Liu, Zhenzhen Xie, Junjie Pang, Chao Cheng. "FedAnchor: Anchored and Adaptive Federated Learning for Fault Diagnosis in Resource-Constrained Industrial IoT." Machines, vol. 14, no. 7, 809, 2026. doi: 10.3390/machines14070809.

[30] Zhaoqi Liu, Ang Cai, Zhen Xu. "Mechanical fault diagnosis under long-tailed data: an improved federated learning method integrating attention, asymmetric convolution, and margin calibration." Scientific Reports, 2026. doi: 10.1038/s41598-026-71247-1.

[31] Genbao Zhao, Juan Zhang. "Relation Knowledge-Guided Federated Model Compression for Rare-Fault Preservation in Motor Fault Diagnosis." Machines, vol. 14, no. 6, 689, 2026. doi: 10.3390/machines14060689.

[32] Shun Liu, Funa Zhou, Shanjie Tang, Xiong Hu, Chaoge Wang, Tianzhen Wang. "Dynamic Semi-Supervised Federated Learning Fault Diagnosis Method Based on an Attention Mechanism." Entropy, vol. 25, no. 10, 1470, 2023. doi: 10.3390/e25101470.

[33] Yichen Jin, Yuqi Luo, Xinyu Liu, Youpeng Fan, Junli Shi. "ProtoFed: Prototype-Enhanced Federated Meta-Learning for Few-Shot Rolling Bearing Fault Diagnosis." Applied Sciences, vol. 16, no. 11, 5277, 2026. doi: 10.3390/app16115277.

[34] Yonghong Zhang, Xingan Xue, Xiaoping Zhao, Lihua Wang. "Federated learning for intelligent fault diagnosis based on similarity collaboration." Measurement Science and Technology, vol. 34, no. 4, 045103, 2023. doi: 10.1088/1361-6501/acab22.

[35] Yan Qin, Ouyang Wang. "Federated fault diagnosis method for collaborative self-diagnosis and cross-robot peer diagnosis." PLOS One, vol. 20, no. 7, e0322484, 2025. doi: 10.1371/journal.pone.0322484.

[36] Zhenhao Yan, Jiachen Sun, Yixiang Zhang, Lilan Liu, Zenggui Gao, Yuxing Chang. "Federated Transfer Learning Strategy: A Novel Cross-Device Fault Diagnosis Method Based on Repaired Data." Sensors, vol. 23, no. 16, 7302, 2023. doi: 10.3390/s23167302.

[37] Guang Yang, Juan Su, Songhuai Du, Qingling Duan. "Federated transfer learning-based distributed fault diagnosis method for rolling bearings." Measurement Science and Technology, vol. 35, no. 12, 126111, 2024. doi: 10.1088/1361-6501/ad76d0.

[38] Tarek Berghout, Mohamed Benbouzid, Toufik Bentrcia, Wei Hong Lim, Yassine Amirat. "Federated Learning for Condition Monitoring of Industrial Processes: A Review on Fault Diagnosis Methods, Challenges, and Prospects." Electronics, vol. 12, no. 1, 158, 2022. doi: 10.3390/electronics12010158.

[39] Weihua Li, Wansheng Yang, Gang Jin, Junbin Chen, Jipu Li, Ruyi Huang, et al.. "Clustering Federated Learning for Bearing Fault Diagnosis in Aerospace Applications with a Self-Attention Mechanism." Aerospace, vol. 9, no. 9, 516, 2022. doi: 10.3390/aerospace9090516.

[40] Narisu Cha, Long Chang. "Addressing Non-IID with Data Quantity Skew in Federated Learning." Information, vol. 16, no. 10, 861, 2025. doi: 10.3390/info16100861.

[41] Mihye Seol, Taejoon Kim. "Performance Enhancement in Federated Learning by Reducing Class Imbalance of Non-IID Data." Sensors, vol. 23, no. 3, 1152, 2023. doi: 10.3390/s23031152.

[42] Guodong Yi, Zhouyang Wu, Xinyu Zhang, Xiaocui Li. "Clustered Federated Learning with Adaptive Similarity for Non-IID Data." Electronics, vol. 14, no. 22, 4454, 2025. doi: 10.3390/electronics14224454.

[43] Sanghui Lee, Jaewook Lee. "Non-IID Degree Aware Adaptive Federated Learning Procedure Selection Scheme for Edge-Enabled IoT Network." Electronics, vol. 14, no. 12, 2331, 2025. doi: 10.3390/electronics14122331.

[44] Qiyun Luo, Qi Tang. "Distribution-Aware Personalised Aggregation for Federated Deep Learning Under Non-IID Data." Applied Sciences, vol. 16, no. 19, 9595, 2026. doi: 10.3390/app16199595.

[45] Qingtie Li, Xuemei Wang, Shougang Ren. "A Privacy Robust Aggregation Method Based on Federated Learning in the IoT." Electronics, vol. 12, no. 13, 2951, 2023. doi: 10.3390/electronics12132951.

[46] Zepei Liu, Zhigang Jin, Zimeng Li, Houyang Luo. "SemLRA: Lightweight Robust Aggregation Against Poisoning Attacks in Semantic Federated Learning." Electronics, vol. 15, no. 20, 4621, 2026. doi: 10.3390/electronics15204621.

[47] Antong Zhou, Ning Jiang, Tong Tang. "Asynchronous Robust Aggregation Method with Privacy Protection for IoV Federated Learning." World Electric Vehicle Journal, vol. 15, no. 1, 18, 2024. doi: 10.3390/wevj15010018.

[48] Haitian Chen, Xuebin Chen, Lulu Peng, Ruikui Ma. "FLRAM: Robust Aggregation Technique for Defense against Byzantine Poisoning Attacks in Federated Learning." Electronics, vol. 12, no. 21, 4463, 2023. doi: 10.3390/electronics12214463.

[49] Emin Akpinar, Murat Taskiran, Bulent Bolat. "FedLucra : Federated Learning With Loss-Utility Controlled Robust Aggregation." Expert Systems, vol. 43, no. 10, e70413, 2026. doi: 10.1111/exsy.70413.

[50] Ibrahim Musa, Shaoxi Teng, Xutong Mu. "FedCWA: Credibility-Weighted Aggregation for Byzantine-Robust Federated Learning." Journal of Networking and Network Applications, vol. 5, no. 2, 77-84, 2025. doi: 10.33969/j-nana.2025.050203.

[51] Zuan Song, Wuzheng Tan, Hailong Wang, Guilong Zhang, Jian Weng. "Dual-Factor Adaptive Robust Aggregation for Secure Federated Learning in IoT Networks." Future Internet, vol. 18, no. 4, 201, 2026. doi: 10.3390/fi18040201.

[52] Cen-Jhih Li, Pin-Han Huang, Yi-Ting Ma, Hung Hung, Su-Yun Huang. "Robust Aggregation for Federated Learning by Minimum γ-Divergence Estimation." Entropy, vol. 24, no. 5, 686, 2022. doi: 10.3390/e24050686.

[53] Jialuo He, Wei Chen, Xiaojin Zhang. "FedAA: A Reinforcement Learning Perspective on Adaptive Aggregation for Fair and Robust Federated Learning." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 16, 17085-17093, 2025. doi: 10.1609/aaai.v39i16.33878.

[54] Xuan Song, Huibin Li, Kailang Hu, Guangjun Zai. "Backdoor Federated Learning by Poisoning Key Parameters." Electronics, vol. 14, no. 1, 129, 2024. doi: 10.3390/electronics14010129.

[55] Chien-Lun Chen, Sara Babakniya, Marco Paolieri, Leana Golubchik. "Defending against Poisoning Backdoor Attacks on Federated Meta-learning." ACM Transactions on Intelligent Systems and Technology, vol. 13, no. 5, 1-25, 2022. doi: 10.1145/3523062.

[56] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, et al.. "Poisoning with Cerberus: Stealthy and Colluded Backdoor Attack against Federated Learning." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, no. 7, 9020-9028, 2023. doi: 10.1609/aaai.v37i7.26083.

[57] Tiandi Ye, Cen Chen, Yinggui Wang, Xiang Li, Ming Gao. "BapFL: You can Backdoor Personalized Federated Learning." ACM Transactions on Knowledge Discovery from Data, vol. 18, no. 7, 1-17, 2024. doi: 10.1145/3649316.

[58] Jiale Zhang, Chengcheng Zhu, Di Wu, Xiaobing Sun, Jianming Yong, Guodong Long. "BADFSS: Backdoor Attacks on Federated Self-Supervised Learning." Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence, 548-558, 2024. doi: 10.24963/ijcai.2024/61.

[59] Wei Shen, Wenke Huang, Guancheng Wan, Mang Ye. "Label-Free Backdoor Attacks in Vertical Federated Learning." Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 19, 20389-20397, 2025. doi: 10.1609/aaai.v39i19.34246.

[60] Georgios Syros, Gokberk Yar, Simona Boboila, Cristina Nita-Rotaru, Alina Oprea. "Backdoor Attacks in Peer-to-Peer Federated Learning." ACM Transactions on Privacy and Security, vol. 28, no. 1, 1-28, 2024. doi: 10.1145/3691633.

AUTHORS

First Author – Sayor Ajfar Aaron, Senior Lecturer, Canadian University of Bangladesh, sayar.aaron@cub.edu.bd.
Second Author – Md. Hasanuzzaman, Chairman, Department of CSE, Canadian University of Bangladesh, hasanuzzaman@cub.edu.bd.
Third Author – Sajjat Hossain Abir, Researcher, Canadian University of Bangladesh, abirsajjathossain@outlook.com.
Third Author – Ashif Newaz, Researcher, Canadian University of Bangladesh, ashifnewaz1@outlook.com.
Fourth Author – Md.Mushfiqur Rahman, Researcher, Canadian University of Bangladesh, mushfiqurrahman0@outlook.com.
Fifth Author – Md Monir Hossain, Researcher, Canadian University of Bangladesh, hossain.monir326@gmail.com.

Correspondence Author – Sayor Ajfar Aaron, Senior Lecturer, Canadian University of Bangladesh, sayar.aaron@cub.edu.bd, +8801684000683.